# 编程

# 写个脚本吧

**题目描述**

**靶场简介**

脚本获取flag

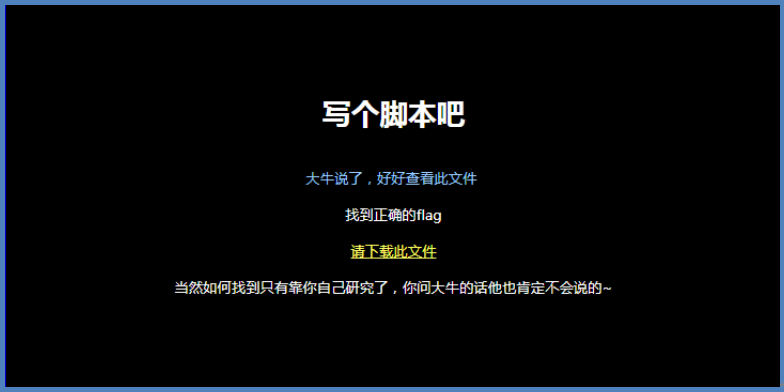
**靶场知识点**

编码解密

**靶场描述**

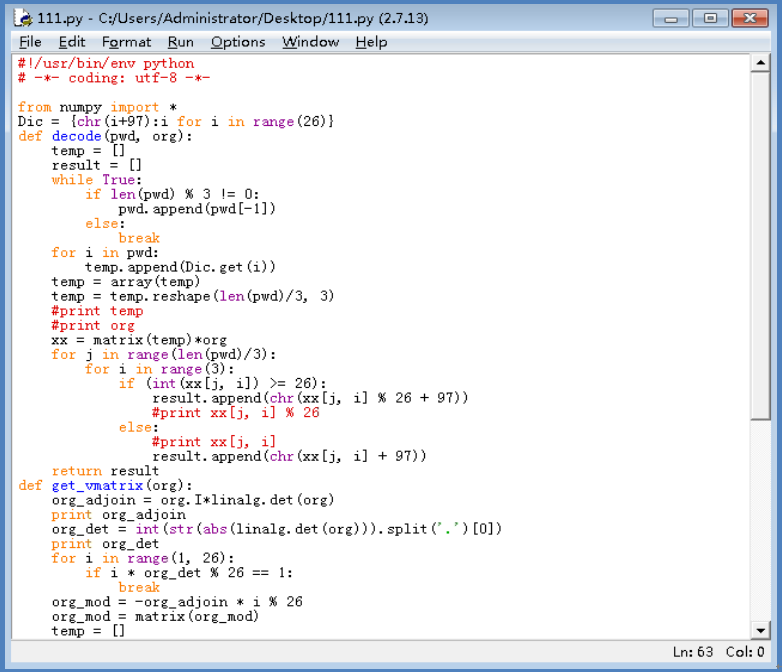
访问网址http://（开启虚拟机后可在此查看）/161,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/161/(IP地址以实际获取的为准)访问如图  


　　步骤2：点击请下载此文件，将文件下载或另存为至电脑任意位置，这里下载到桌面。  


　　步骤3：双击打开刚刚下载的文本文档，可看到题目信息。  


　　步骤4：编写脚本，导出flag值  


　　脚本：

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

from numpy import \*

Dic = {chr(i+97):i for i in range(26)}

def decode(pwd, org):

temp = []

result = []

while True:

if len(pwd) % 3 != 0:

pwd.append(pwd[-1])

else:

break

for i in pwd:

temp.append(Dic.get(i))

temp = array(temp)

temp = temp.reshape(len(pwd)/3, 3)

#print temp

#print org

xx = matrix(temp)\*org

for j in range(len(pwd)/3):

for i in range(3):

if (int(xx[j, i]) >= 26):

result.append(chr(xx[j, i] % 26 + 97))

#print xx[j, i] % 26

else:

#print xx[j, i]

result.append(chr(xx[j, i] + 97))

return result

def get\_vmatrix(org):

org\_adjoin = org.I\*linalg.det(org)

print org\_adjoin

org\_det = int(str(abs(linalg.det(org))).split('.')[0])

print org\_det

for i in range(1, 26):

if i \* org\_det % 26 == 1:

break

org\_mod = -org\_adjoin \* i % 26

org\_mod = matrix(org\_mod)

temp = []

for i in range(org\_mod.shape[0]):

for j in range(org\_mod.shape[1]):

temp.append(int(str(org\_mod[i, j]).split('.')[0]))

org\_final = matrix(temp).reshape(org\_mod.shape[0], org\_mod.shape[1])

#print org\_final

return org\_final

if \_\_name\_\_ == '\_\_main\_\_':

''' for test

pwd = list("act")

org = matrix(array([[6, 24, 1], [13 , 16, 10], [20, 17, 15]]))

result = decode(pwd, org)

print "".join(result)

deorg = matrix(array([[8, 5, 10], [21 , 8, 21], [21, 12, 8]]))

result = decode(result, deorg)

print "".join(result)

'''

pwd = "wjamdbkdeibr"

pwd = list(pwd)

org = matrix(array([[1,2,3],[4,5,6],[7,8,10]]))

org\_vm = get\_vmatrix(org)

print org\_vm

print " flag :" + "".join(decode(pwd, org\_vm))

　　步骤5：输出flag值

## btsyjtlvsslb

# key

**题目描述**

**靶场简介**

分析加密算法，写出注册机

**靶场知识点**

编程解密

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/282,找到其中的key或者flag

**Write-up**

　　分析加密算法，写出注册机，脚本如下

#Solution

key = "themidathemidathemida"

brainf = ">----++++....<<<<."

initial\_key =""

for i in range(18):

initial\_key += chr((ord(key[i]) ^ ord(brainf[i]))+22)

print "Initial key:",initial\_key

secret\_key = "".join([chr(ord(i)+9) for i in initial\_key])

print "Secret:",secret\_key

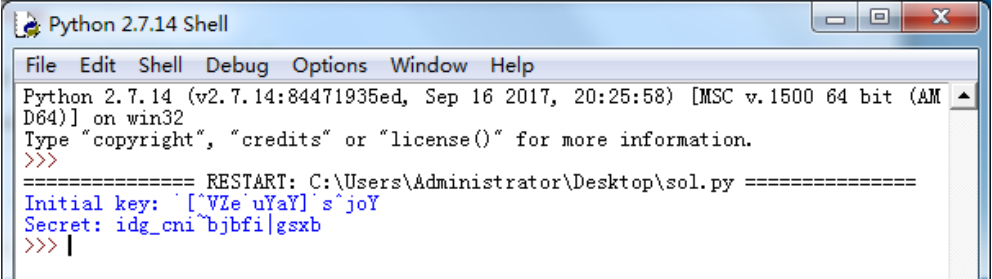
"""

f = open("C:\Users\CSAW2016\haha\flag\_dir\flag.txt")

f.write(secret\_key)

f.close()

"""



Initial key: `[^VZe`uYaY]`s^joY

## idg\_cni~bjbfi|gsxb

# 寻找明文

**题目描述**

**靶场简介**

脚本导出flag

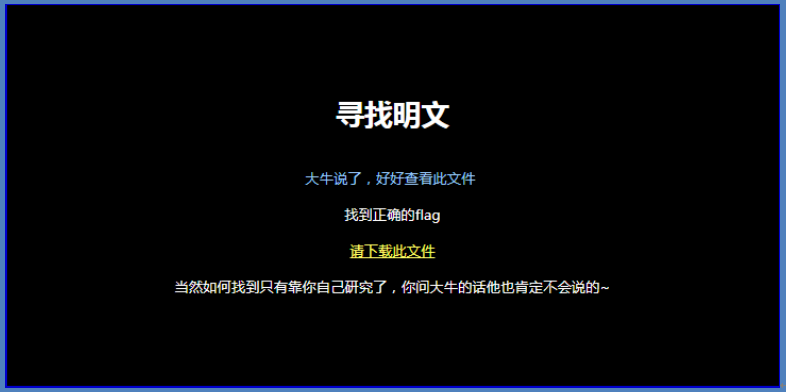
**靶场知识点**

编码解密

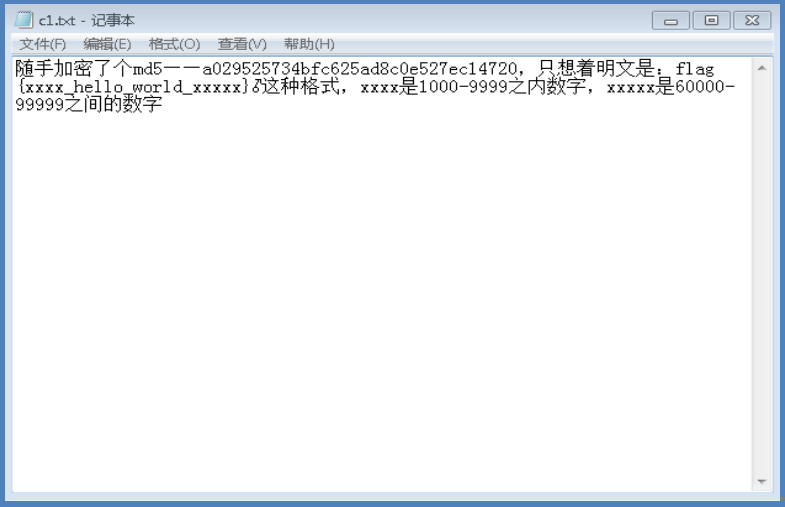
**靶场描述**

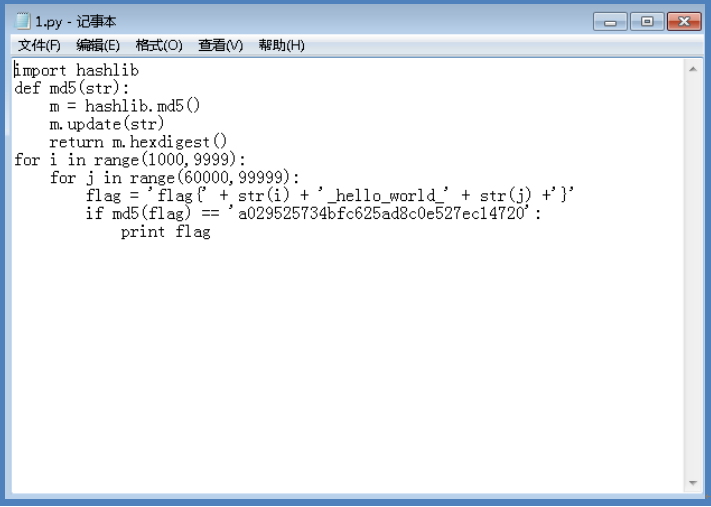
访问网址http://10.2.0.79/166,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/166/ (IP地址以实际获取的为准)访问如图  


　　步骤2：点击请下载此文件，将文件下载至任意位置，这里下载到桌面。  


　　步骤3：可以看到文件是一个txt文本。双击打开文本，可看到flag信息。  


　　步骤4：根据之前类似题目及相关了解，需用脚本导出flag，新建1.py，写入脚本并执行。  


　　脚本：

import hashlib

def md5(str):

m = hashlib.md5()

m.update(str)

return m.hexdigest()

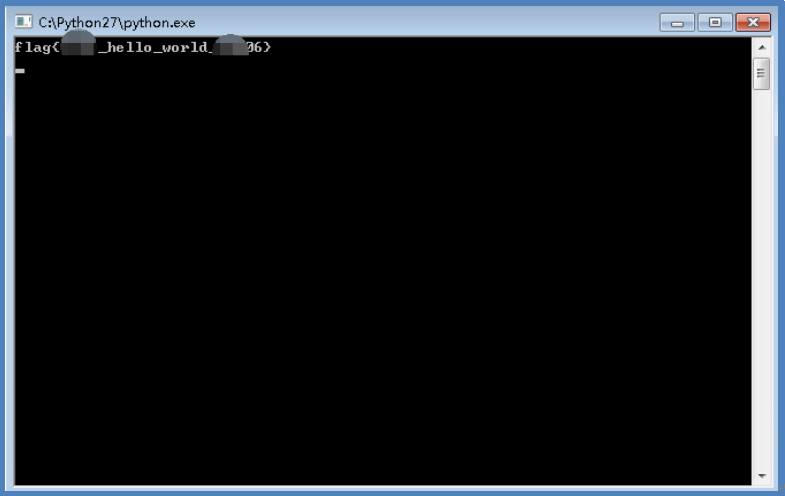
for i in range(1000,9999):

for j in range(60000,99999):

flag = 'flag{' + str(i) + '\_hello\_world\_' + str(j) +'}'

if md5(flag) == 'a029525734bfc625ad8c0e527ec14720':

print flag

　　步骤5：运行脚本得出flag  


## flag{1121\_hello\_world\_60006}

# vbs

**题目描述**

**靶场简介**

VBS编码修改且更换计算方式

**靶场知识点**

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/16,找到其中的key或者flag。  
VBS文件修改编码且更换计算方式的方式获取flag。

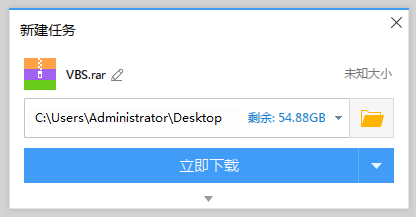
**Write-up**

修改vbs程序，输出核心代码，分析解密算法，最终反推出答案

**步骤1：**打开浏览器输入网址<http://10.10.23.105/16/>(**IP地址以实际获取的为准**) 访问如图

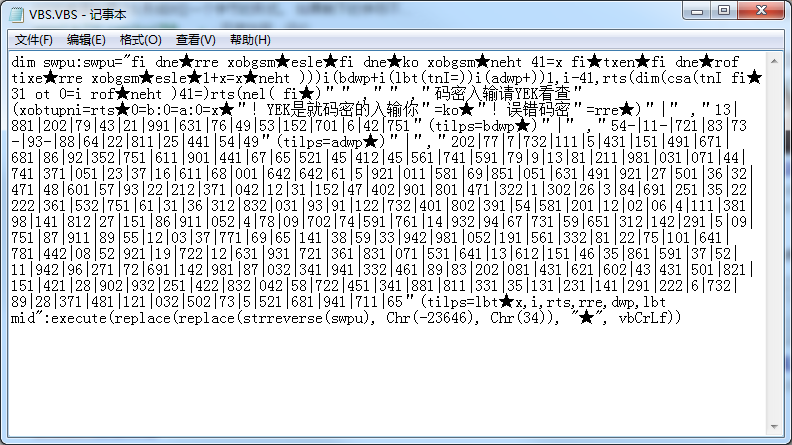


**步骤2：**点击下载，将rar文件下载



**步骤3：**下载题目文件，解压出题目文件VBS.VBS，然后用记事本打开，注意末尾的代码：

:execute(replace(replace(strreverse(swpu), Chr(-23646), Chr(34)), "★", vbCrLf))



**步骤4：**显然execute后的括号里是在进行代码解密还原，我们现在需要明文代码，将末尾代码修改为：

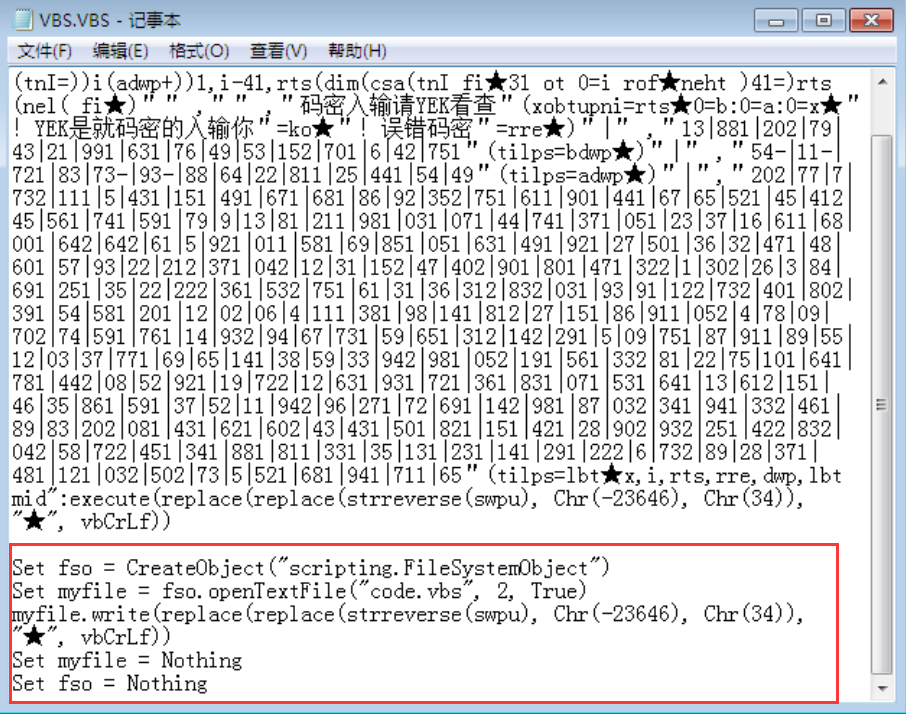
Set fso = CreateObject("scripting.FileSystemObject")

Set myfile = fso.openTextFile("code.vbs", 2, True)

myfile.write(replace(replace(strreverse(swpu), Chr(-23646), Chr(34)), "★",  
vbCrLf))

Set myfile = Nothing

Set fso = Nothing



**步骤5：**修改后保存，然后打开输出文件code.vbs，分析代码，发现关键算法如下：

* 文件路径：C:\Windows\System32

if (len(str)=14) then

for i=0 to 13

if Int(asc(mid(str,14-i,1))+pwda(i))=Int(tbl(i+pwdb(i))) then

x=x+1

else

msgbox err

exit for

end if

next

if x=14 then

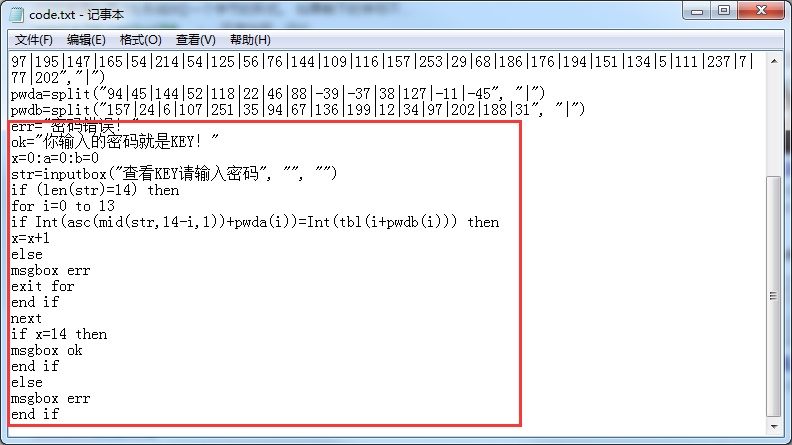
msgbox ok

end if

else

msgbox err

end if



**步骤6：**首先，输入的文本长度必须为14，接着就是每一位的验证：

Int(asc(mid(str,14-i,1))+pwda(i))=Int(tbl(i+pwdb(i)))

只有满足这个条件的字母，程序才会继续验证下一条，否则就报错，分析一下这句判断，pwda、pwdb、tbl都是常量数组，因此这里只需要进行反向计算即可。

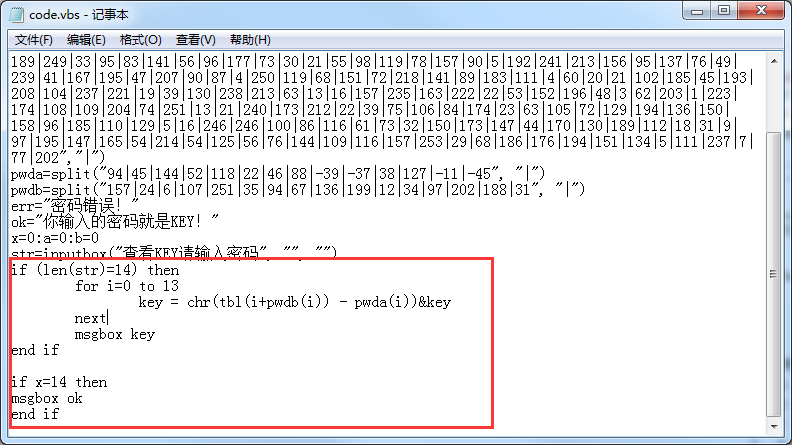
将循环部分的代码改为如下代码：

for i=0 to 13

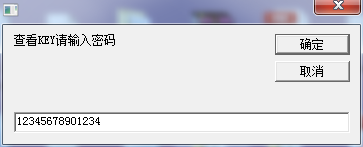
key = chr(tbl(i+pwdb(i)) - pwda(i))&key'验证是倒序的，所以这里也应该倒序

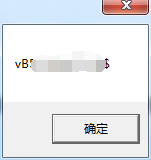
next

msgbox key



**步骤7：**再次运行code.vbs这个VBS，要输14位数，因为一开始有个判断，点击确定，即可得到本题的KEY。





# 小黑的语言

**题目描述**

**靶场简介**

编程修改内容，是二维码复原

**靶场知识点**

编程转换

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/20,找到其中的key或者flag。  
通过内容发现隐藏内容，再通过编程获取隐藏信息。

**Write-up**

**步骤1：**打开浏览器输入网址<http://10.10.23.105/20>(**IP地址以实际获取的为准**)访问如图

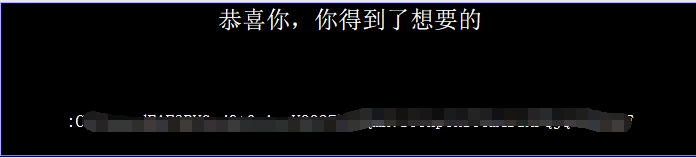


**步骤2：**细心的朋友应该可以看出这是一个二维码，只是间隔太大扫不出来，把内容复制下来，通过编程修改成可以扫的。





**步骤3：**将二维码的内容提交即可，得到答案。



# 汇编QAQ

**题目描述**

**靶场简介**

解析机器码内容

**靶场知识点**

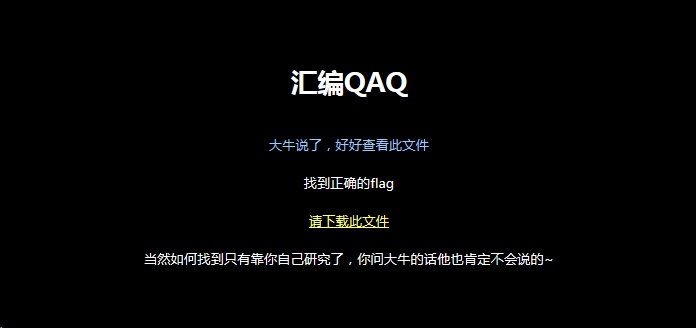
编程转换

**靶场描述**

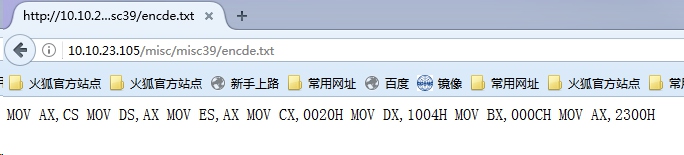
访问网址http://（开启虚拟机后可在此查看）/78，通过获取内容，判断内容形式，使用机器码解码，找到其中的key或者flag。

**Write-up**

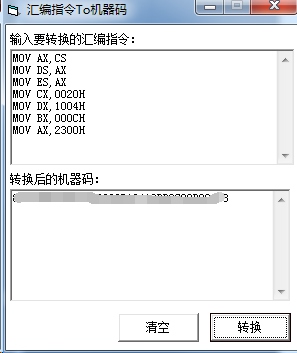
步骤1：打开浏览器输入网址<http://10.10.23.105/78/>（**IP地址以实际获取为准**）访问如图。



步骤2：点击 请下载此文件 跳转文本文件。



步骤3：使用汇编转换成机器码的软件：AsmToHex，就是flag。工具路径：\Desktop\ctf\_tools\asmtohex(汇编转换成机器码).zip



# 小py

**题目描述**

**靶场简介**

操作机用户名root，密码123456 完成加密算法

**靶场知识点**

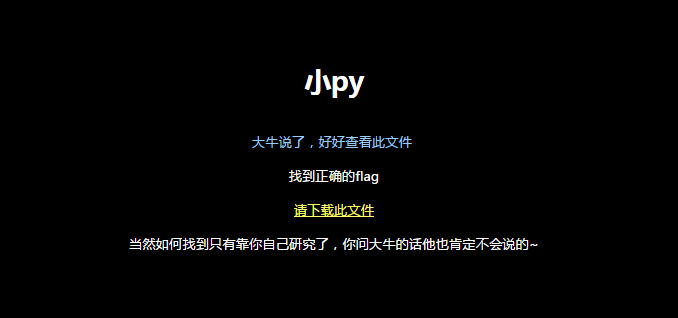
编程解密

**靶场描述**

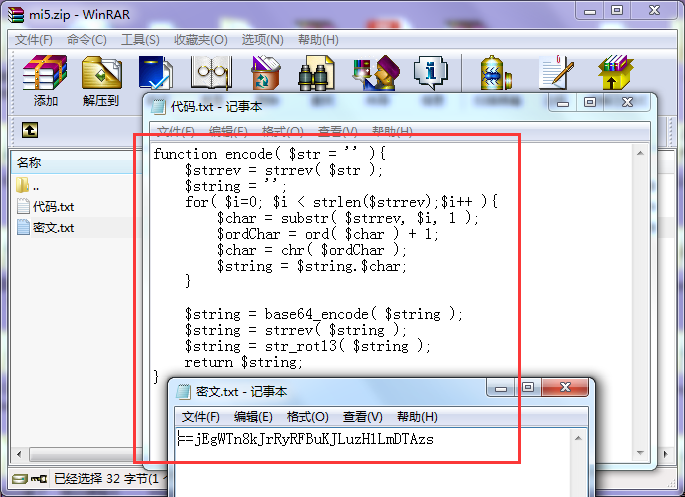
访问网址http://（开启虚拟机后可在此查看）/88,找到其中的key或者flag

**Write-up**

**步骤1：**打开浏览器输入网址<http://10.10.23.105/88/>（**IP地址以实际获取的为准**）访问如图



**步骤2：**点击 请下载此文件 是一个压缩包，打开后发现，代码和密文



**步骤3：**通过加密算法，反推解密算法

<?php

function decode($string){

$string = str\_rot13($string);

$string = strrev($string);

$string = base64\_decode( $string );

$str = '';

for( $i=0; $i < strlen($string);$i++ ){

$char = substr( $string, $i, 1 );

$ordChar = ord( $char ) - 1;

$char = chr( $ordChar );

$str = $str.$char;

}

$str = strrev($str);

return $str;

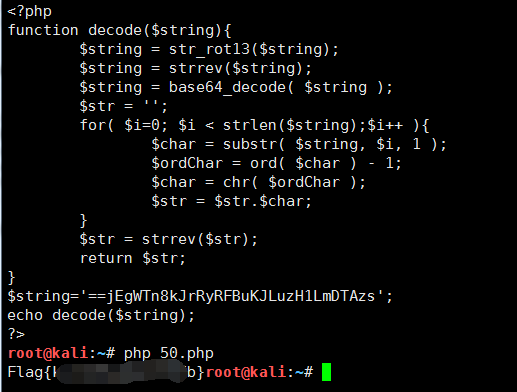
}

$string='==jEgWTn8kJrRyRFBuKJLuzH1LmDTAzs';

echo decode($string);

?>

**步骤4：**运行解密算法，得到flag



## Flag{kxCHGMwXWgQ45BEb}

# Babyre

**题目描述**

**靶场简介**

mips题目，简单的base64变形解码，替换了base64置换表

**靶场知识点**

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）:20000，通过web获取的文件，使用相应工具获取文件中隐藏的内容，得到flag

**Write-up**

执行脚本：

#! / usr / bin / env

#-- coding:utf - 8 - -

base64list = 'R9Ly6NoJvsIPnWhETYtHe4Sdl+MbGujaZpk102wKCr7/0Dg5zXAFqQfxBicV3m8U'

cipherlist = "eQ4y46+VufZzdFNFdx0zudsa+yY0+J2m"

length = len(cipherlist)

print length

group = length / 4

s = ''

string = ''

for i in range(group - 1):

j = i\*4

s = cipherlist[j:j + 4]

string += chr(((base64list.index(s[0])) << 2) + ((base64list.index(s[1])) >> 4))

string += chr(((base64list.index(s[1]) & 0x0f) << 4) + ((base64list.index(s[2])) >> 2))

string += chr(((base64list.index(s[2]) & 0x03) << 6) + ((base64list.index(s[3]))))

j = (group - 1)\*4

s = cipherlist[j:j + 4]

string += chr(((base64list.index(s[0])) << 2) + ((base64list.index(s[1])) >> 4))

if s[2] == '=':

print string

else:

string += chr(((base64list.index(s[1]) & 0x0f) << 4) + ((base64list.index(s[2])) >> 2))

if s[3] == '=':

print string

else:

string += chr(((base64list.index(s[2]) & 0x03) << 6) + ((base64list.index(s[3]))))

print string

　　得到flag值  
SUCTF{wh0\_1s\_y0ur\_d4ddy}

## SUCTF{wh0\_1s\_y0ur\_d4ddy}

# ！冒泡吧

**题目描述**

**靶场简介**

排序算法推算内容

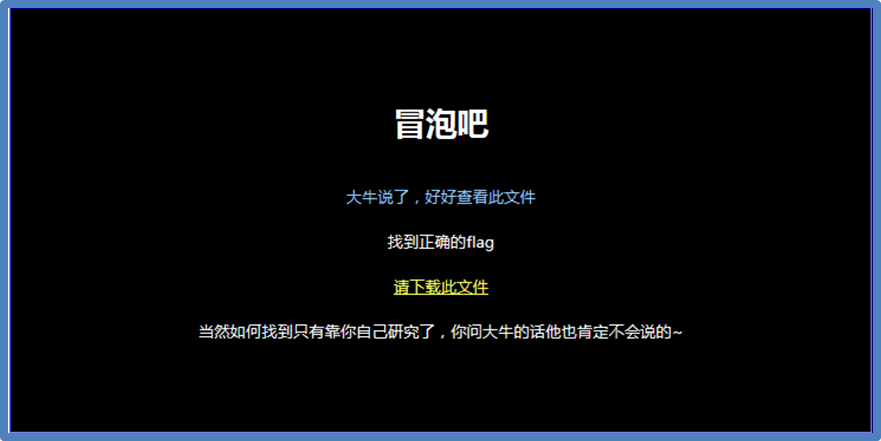
**靶场知识点**

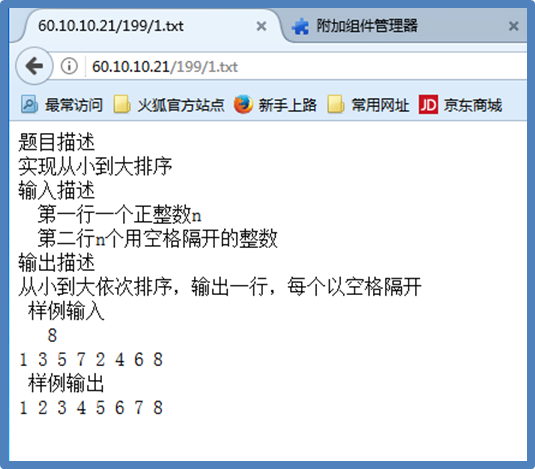
编码解密

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/199，通过web获取的信息，使用相应的方法，得到排序后的内容推算出flag。

**Write-up**

步骤1：打开浏览器输入网址http://10.10.23.105/199/ (IP地址以实际获取的为准)访问如图。  
  
步骤2：本题考查的是对算法的理解，点击下载文件，可以看到题目。

  
步骤3：编写一个排序算法，将给出的数进行排序依照题目要求得出flag。

#include <stdio.h>

#include <string.h>

void Swap(int a[],int length)

{

int i,j,temp;

for(i=0;i<length;i++)

{

for(j=length-1;j>i;j--)

{

if(a[j]<a[j-1])

{

temp=a[j];

a[j]=a[j-1];

a[j-1]=temp;

}

}

}

}

int main()

{

int i;

int n;

int a[100];

scanf("%d",&n);

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

Swap(a,n);

for(i=0;i<n;i++)

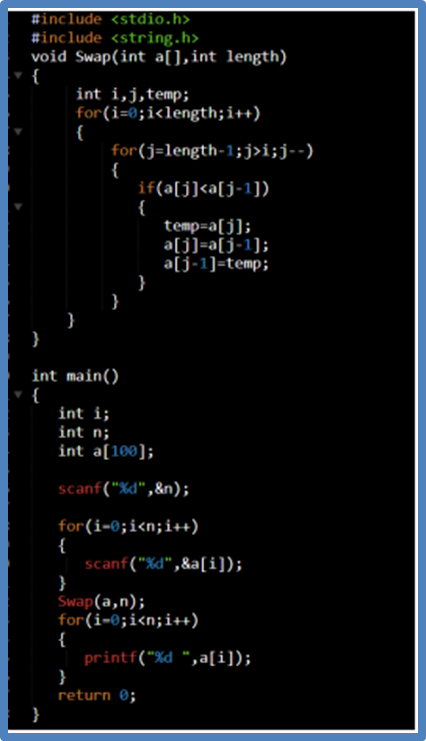
{

printf("%d ",a[i]);

}

return 0;

}

  
步骤4： 用编写出的程序将8 1 3 5 7 2 4 6 8 ，5 6 7 3 6 7 4 5 8，5 8 3 2 0 6 7 3 7，三段数据从大到小排序，找到每组数中的第一个数，倒数第一个数，倒数第三个数。将三个组的数按照第一组、第二组、第三组的顺序放在一起得到本题目flag。  
12345678  
36677  
02368

## 123456783667702368

# ！斐波那契数列

**题目描述**

**靶场简介**

斐波那契数列的定义算法

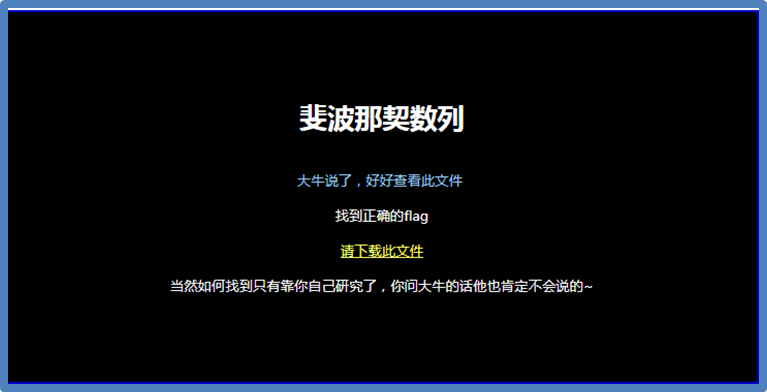
**靶场知识点**

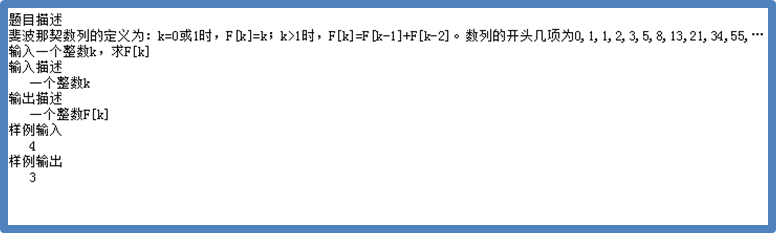
编码解密

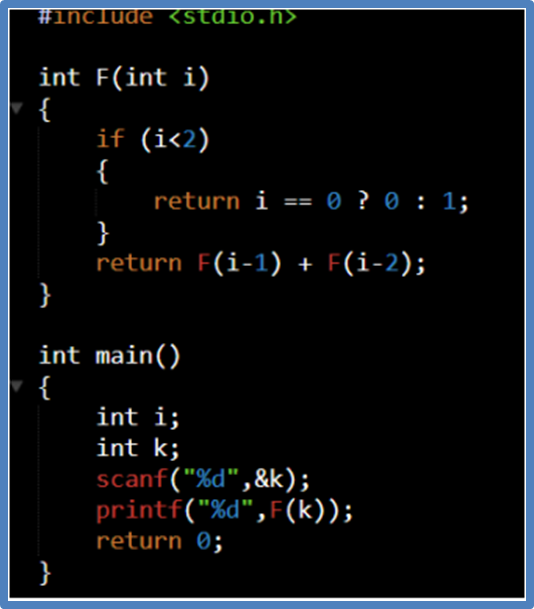
**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/200，通过web获取的内容，使用相应的算法，计算信息内容，得到flag。

**Write-up**

步骤1：打开浏览器输入网址http://10.10.23.105/200/ (IP地址以实际获取的为准)访问如图。  
  
步骤2：点击下载文件，可以看到题目描述。

  
步骤3：根据题目描述编写以下代码。



步骤4：依次使用程序将下面的每个k解出对应的F[k]的值，得到本题的flag。  
K={2,3,1,5,1,8,0,5,5,0,1,0,2,3,4}  
将F[2],F[3],F[1],F[5],F[8],F[0],F[5],F[5],F[0],F[1],F[0],F[2],F[3],F[4]。所有的值依次计算出来，得到本题的flag。



## 1215121055010123

# 最短路径

**题目描述**

**靶场简介**

弗洛伊德算法

**靶场知识点**

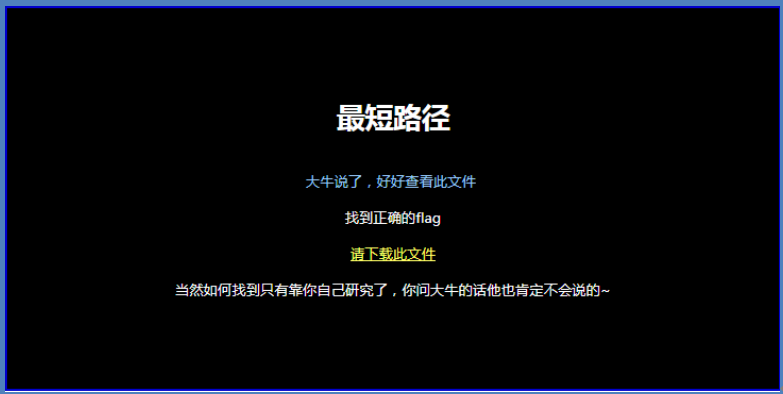
编码解密

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/201,找到其中的key或者flag

**Write-up**

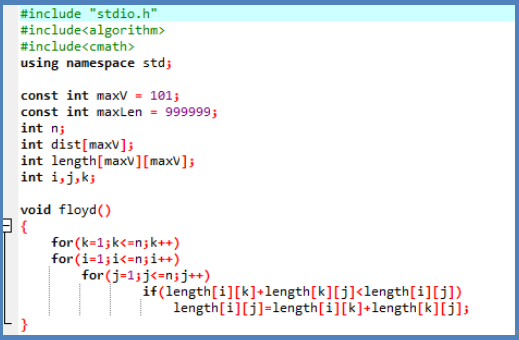
　　步骤1：打开浏览器输入网址http://10.10.23.105/201/ (IP地址以实际获取的为准)访问如图



　　步骤2：点击下载文件，可以看到本题的题目  
　　题目：本题考查对弗洛伊德算法的应用，要求自行设计算法，从四个顶点四条边的有向图中，输出顶点1开始到每个顶点的距离，将对应的数值进行排列可以得到本题的flag。



　　步骤3：编写一个弗洛伊德算法程序



# 黄金分割法

**题目描述**

**靶场简介**

黄金分割法获取flag

**靶场知识点**

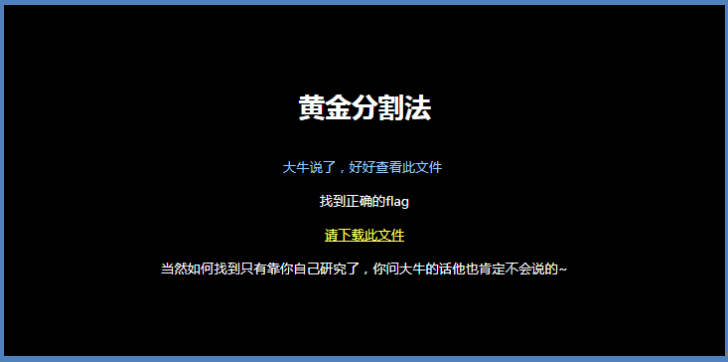
编码解密

**靶场描述**

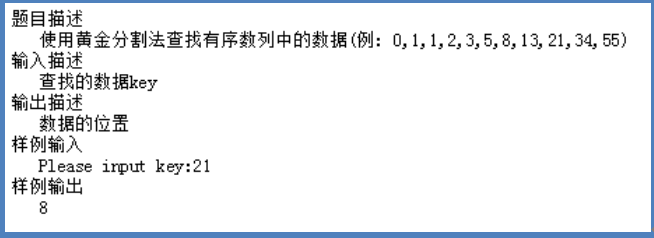
访问网址http://（开启虚拟机后可在此查看）/202,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/202/ (IP地址以实际获取的为准)访问如图

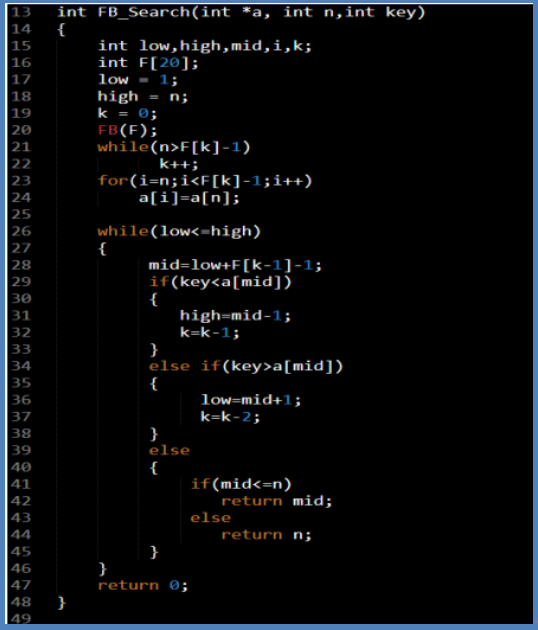


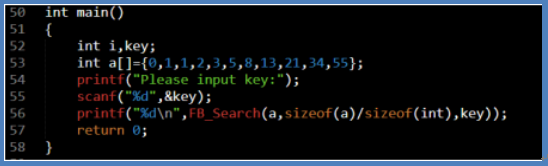
　　步骤2：点击下载文件，可以得到题目信息



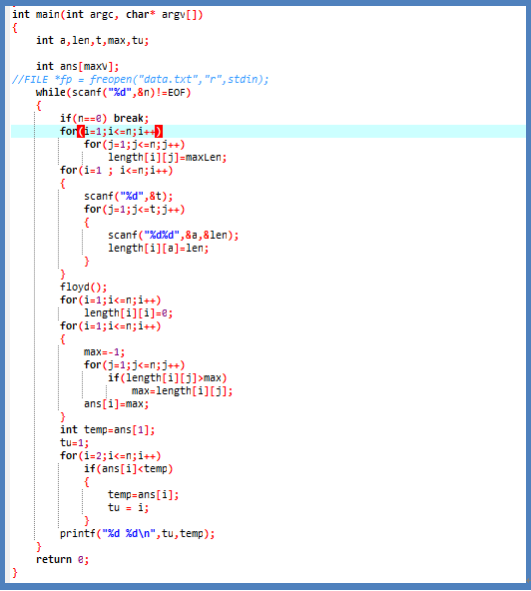
　　步骤3：设计一个黄金分割法的程序







　　步骤4：依次输出1,2,8,1,0,1,3,3,5,8,2,3,0,5,2,3,2,1,0,1在数列中的序号，得到本题的flag。



步骤4：解出每个有向图中顶点1到各个顶点的值，并将其按照从大到小的顺序排列，得到对应字母的值，排列得出flag

# KPM

**题目描述**

**靶场简介**

KPM模式匹配算法

**靶场知识点**

编码解密

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/203,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/203/ (IP地址以实际获取的为准)访问如图  

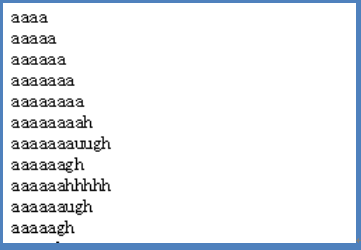

　　步骤2：点击下载文件，可以得到题目信息

统计key.txt中flag的总个数。

要求：1.C语言或C++

2.使用KMP模式匹配算法或其他有效算法

　　步骤3：点击下载文件2，可以看到一个字典key.txt



　　步骤4：编写KVM算法来对flag进行破解。





　　步骤5：将字典key.txt与程序放置同一目录下，执行程序对flag进行破解，可以得到本题flag。



# 很简单吧

**题目描述**

**靶场简介**

base64解密

**靶场知识点**

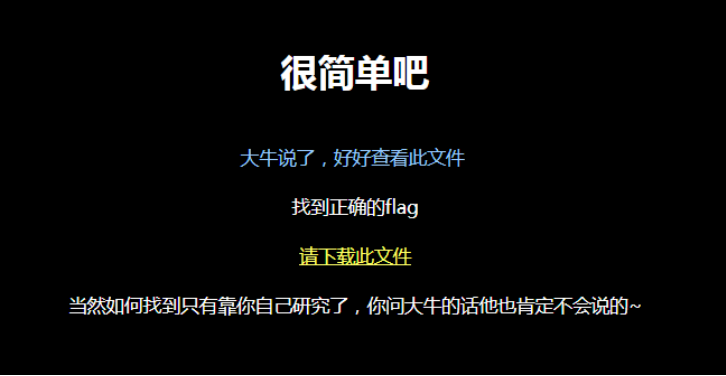
编码解密

**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/204,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/204/ (IP地址以实际获取的为准)访问如图



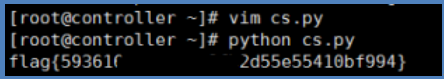
　　步骤2：点击下载文件，是一个记事本里面有base64的代码



　　步骤3：将代码处理出来，就可以得到我们的flag。  
　　编写python程序，在画线处粘贴base64代码（需要加单引号）。



　　步骤4：保存退出，执行命令程序得到本题flag



# ！算法crc32

**题目描述**

**靶场简介**

crc32解密

**靶场知识点**

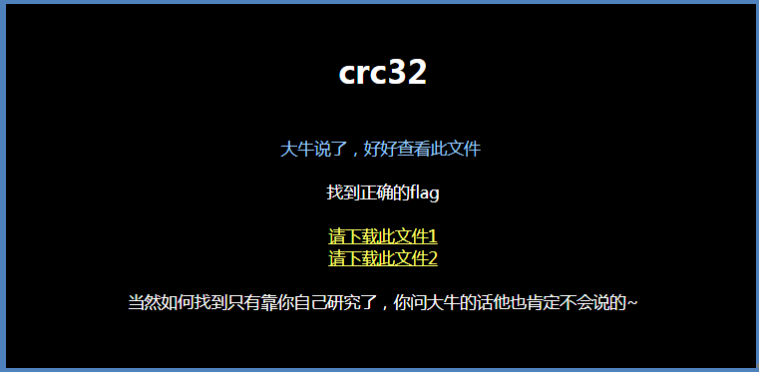
编码解密

**靶场描述**

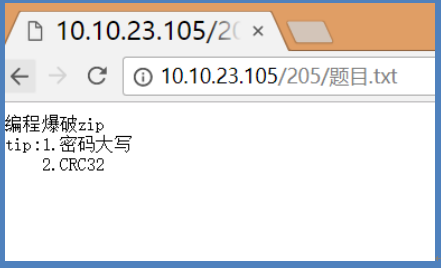
访问网址http://（开启虚拟机后可在此查看）/205,找到其中的key或者flag

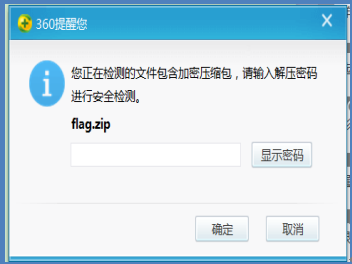
**Write-up**

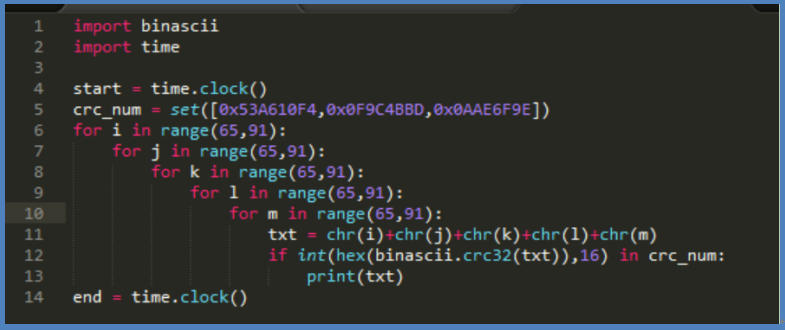
　　步骤1：打开浏览器输入网址http://10.10.23.105/205/ (IP地址以实际获取的为准)访问如图



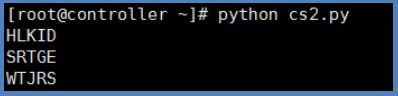
　　步骤2：打开下载文件1，和下载文件2得到跟flag有关的信息，需要输入解压密码才能得到flag。

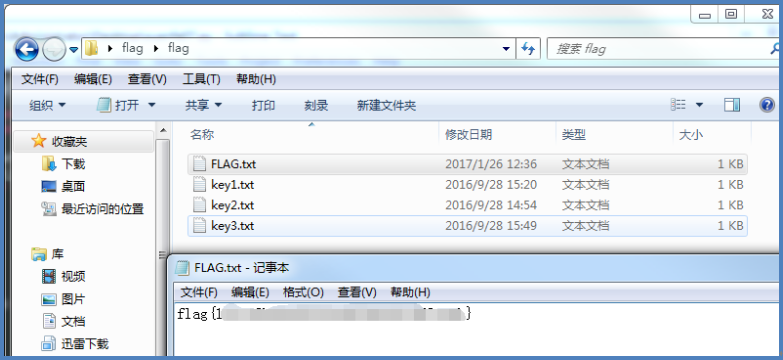




　　步骤3：由题目可知，需要得到flag压缩包的解压密码  
　　编写python程序  


　　步骤4：运行python程序求得密码。将密码顺序重新排列一下得到密码HLKIDWTJRSSRTGE，解开flag压缩包，得到本题flag。





## flag{19c122b00447f21a2938e2602d53ceab}

# MD5

**题目描述**

**靶场简介**

编程爆破

**靶场知识点**

编码解密

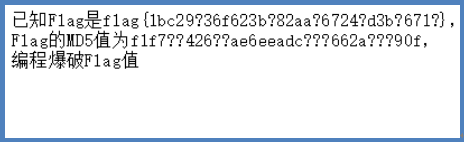
**靶场描述**

访问网址http://（开启虚拟机后可在此查看）/206,找到其中的key或者flag

**Write-up**

　　步骤1：打开浏览器输入网址http://10.10.23.105/206/ (IP地址以实际获取的为准)访问如图

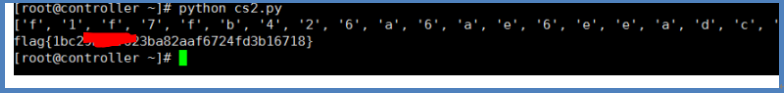


　　步骤2：点击下载文件可以看到此题的题目描述  


　　步骤3：编写python程序对其进行破解。创建一个记事本pojie.txt,



　　步骤4.运行程序得到flag



## flag{1bc29b36f623ba82aaf6724fd3b16718}