Hgame Week1 Writeup

WEB

0x01 谁吃了我的 flag

一看是第一题，f12+ctrl u，啥都没有发现，然后左点点右点点还是没有任何收获，陷入沉思，直到 hint 放出来时，我还是不会做 Orz

来看 hint: 那个夜晚他正在用vim编写题目页面，似乎没有保存就关机

搜索学习一波

一、vim备份文件

默认情况下使用Vim编程，在修改文件后系统会自动生成一个带~的备份文件，某些情况下可以对其下载进行查看；

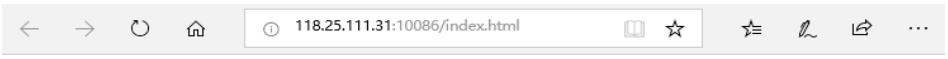
eg:index.php普遍意义上的首页，输入域名不一定会显示。 它的备份文件则为index.php~

二、vim临时文件

vim中的swp即swap文件，在编辑文件时产生，它是隐藏文件，如果原文件名是submit，则它的临时文件

.submit.swp。如果文件正常退出，则此文件自动删除。

果然如此 118.25.111.31:10086/.index.html.swp



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_diScl0Sure

flag get!!!

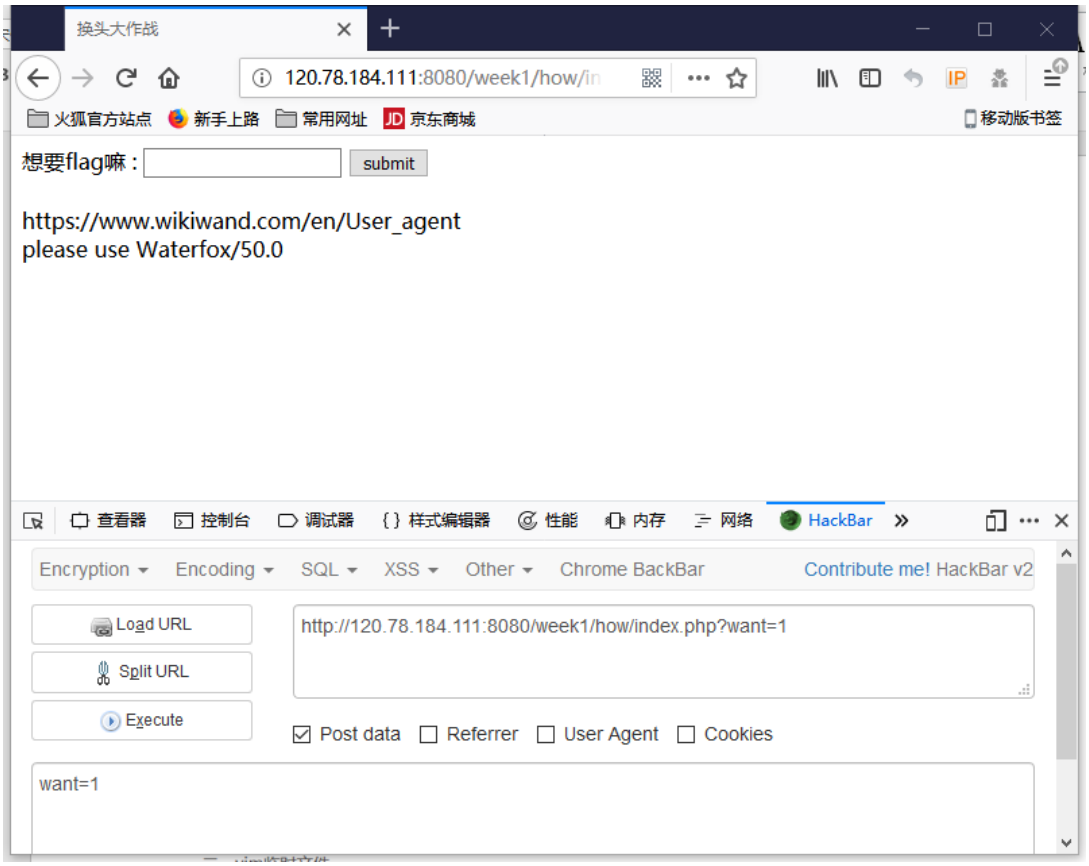
0x02 换头大作战

想要flag嘛：[ ] submit

request method is error.I think POST is better

先随便输个东西进去

用工具 hackbar post 数据过去



然后按照提示一路改过去，注意：改 User Agent 的时候要 please use 也加上，我一开始没加，太坑了。。。

想要flag嘛：[ ] submit

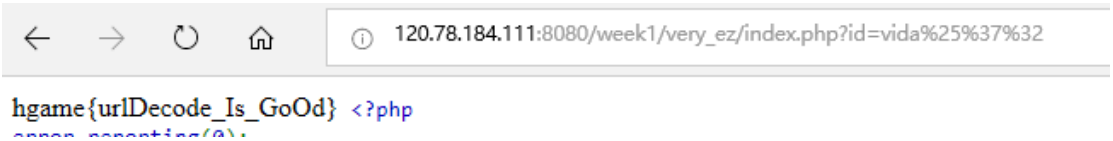
hgame{hTTp_HeaDeR_iS_Ez}

0x03 代码审计初🕯体验

```
urldecode($_GET['id']);
```
urldecode 二次编码绕过

第一次 r->%72 第二次%72->%25%37%32 注：只编码一个字母就可以

flag get !!



0x04 为什么不问问神奇的十二姑娘和她的小伙伴呢

首先，万能的 F12+ctrl u

```
<a href="f12.php"></a>
```

接着使用 Wireshark 抓包

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Thu, 31 Jan 2019 09:31:52 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/7.2.14
password: woyaoflag
```

Password:woyaoflag

使用 Hackbar post 过去，然后提示click me to get flag

继续使用 Wireshark 进行抓包 flag get!!!

```
<html>
        <head>
                <title>can you find me?</title>
        </head>
        <body>
                <p>flag:hgame{f12_1s_aMazIng111}</p>
        </body>
</html>
```

RE

0x01 brainfxxker

一开始以为直接找个 brainf**k 的解释器跑一下就行，然而发现事情并不简单 Orz,整串 brainf**k 分为 9 个部分，取其中一部分来看

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

10 个加号，10 个减号，后面又有 2 个加号，反复分析，发现了一个惊天大秘密!!

其效果等同于 a*b±c

于是问题就简单了

```
  parser.execute(",>10[<10>-]<-2[+.],>9[<9>-]<+1[+.],7[<7>-]<+3[+.],
}98 82 52 33 78 102  85 99 75
bR4!NfUcK
```

直接手撸!!!

0x02 HelloRe

```
unsigned __int64 v8; // [rsp+28h] [rbp+0h]

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;
```

IDA 打开，flag get

0x03 Pro 的 Python 教室(一)

```
enc2 = 'SGVyZV8xc18zYXN5Xw=='

secend = raw_input()
secend = base64.b64encode(secend)
if secend == enc2:
```

将第二段字符串 base64 解码一下 flag 就出来
hgame{Here_1s_3asy_Pyth0n}

PWN

0x01 aaaaaaaaaa

nc 连上输入超过 99 个 a 就可以拿到 shell

```
akumu@akumu-virtual-machine:~$ nc 118.24.3.214 9999
Welcome to PWN'world!let us aaaaaaaaaa!!!
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
cat flag
hgame{Aa4_4aA_4a4aAAA}
```

0x02 薯片拯救世界 1

```
puts(&byte_80489A4);
getchar();
while ( 1 )
{
  puts(&byte_8048A18);
  read(0, &buf, 0x18u);
  n = strlen(&buf);
  if ( !strncmp(s1, &buf, n) )
    break;
  puts(asc_8048A58);
}
puts(asc_8048A80);
getchar();
return 0;
```

先扔进 ida 里看看，万能的 f5，看到 read 函数，考虑栈溢出，然后并不是，不会做 orz，hint 出来：可以通过爆破一点点得到 flag

百度一下 strncmp 函数，发现它是逐位比较的，第三个参数 n 也就是我们输入的字符串的长度，也就是说，我们可以一位位爆破出 flag，python 苦手写了一万年终于写出来了脚本 orz，上脚本：

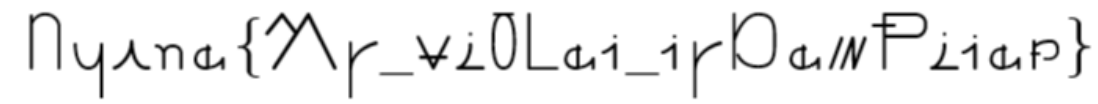
```
#!/usr/bin/python
from pwn import*
password=''
for i in range(24):
    c=remote('118.24.3.214',10001)
    c.send('\n')
    print(c.recv())
    c.send('\n')
    print(c.recv())
    c.send('\n')
    print(c.recv())
    c.send('\n')
    print(c.recv())
    c.send('\n')
    print(c.recv())
    for j in range(32,127):
        try:
            c.send(password+chr(j)+'\0') #天坑，read 函数只有读到\0 才结束
            print(password+chr(j))
            print(c.recv())
        except:
            password+=chr(j-2) #不知道为什么，每次总是多循环了 2 次，神奇
            break
    c.close()
print(password)
```

MISC

0x01 Hidden Image in LSB

直接上工具 stegsolve

0x02 打字机



紫罗兰永恒花园里的打字机 Orz,

对照着翻译，注意大小写

hgame{My_viOlet_tyPewRiter}

0x03 Broken Chest

https://ctf-wiki.github.io/ctf-wiki/misc/archive/zip/

先看一波 hint 里的资料，zip 文件头标识由固定值 50 4B 03 04 组成，打开压缩包发现损坏，使用 winhex 打开，开头的 49 改成 50

Offset	0	1	2	3	4
00000000	50	4B	03	04	14
00000016	B3	B0	22	00	00

发现压缩包可以打开了，发现需要密码，

```
  X      S0mEThIn
g_U5efuL
```

密码在结尾处，打开就发现了 flag，hgame{Cra2y_DIaMOnd}

0x04 Try

流量分析题，Wireshark 打开，发现其中的一个

```
.......dec/password.txthgame********PK..?........d8N.............
$..............dec/
. .........~.......~.......Qy......PK..?.
......c8N....*Q..*Q....$.......  ..."...dec/open-it.zip
```

http 包很可疑，导出 http 的这个包，发现是个压缩包文件，解压，又出现了一个压缩包文件，需要密码，其中有个 txt 文件提示了密码的前 5 位 hgame，而后紧跟 8 个*，推测后面是 8 位数字密码，使用字典进行字典攻击，1 分半就跑出来了，跑出来的密码是 hgame25839421，你以为这就结束了吗？打开后发现是我老婆的照片

orz，我老婆真漂亮，真漂亮，真漂亮，真漂亮，使用 winhex 打开，发现 pk 字样，怀疑是 zip 文件，改后缀，打开发现又需要密码，这次没有任何提示，怀疑是伪加密，直接利用 zip 自带功能修复一下，可以直接打开了，解压出来是个 docx 文件，打开一片空白，继续用 winhex 打开，又是压缩包文件，orz，修改后缀，在一堆 xml 文件中，打开 document.xml 文件，终于，我们发现了 flag 标志的字样，一开始我以为那串 59 开头的字符串还要解密，然而捣鼓了半天什么都没解出来，太坑了，直接提交，发现正确了，orz，flag

```
øt2NÝÄÍ"ç$
'/    $
      1.docx
        Æ,Ÿšø&Ô
·;"fœ³Ô J7Ý ø&Ô
PK          X
 %
```

```
<w:t>hgame</w:t>
</w:r>
- <w:r w:rsidRPr="004066DE">
  - <w:rPr>
      <w:rFonts w:hint="eastAsia"/>
      <w:vanish/>
    </w:rPr>
    <w:t>{</w:t>
</w:r>
- <w:r w:rsidRPr="004066DE">
  - <w:rPr>
      <w:vanish/>
    </w:rPr>
    <w:t>59d28413e36019861498e823f3f41406</w:t>
</w:r>
- <w:r w:rsidRPr="004066DE">
  - <w:rPr>
      <w:rFonts w:hint="eastAsia"/>
      <w:vanish/>
    </w:rPr>
    <w:t>}</w:t>
```

hgame{59d28413e36019861498e823f3f41406}

CRYPTO

0x01 Mix

--...../....-/....-/-.../--.../...--/...../..-./-..../-../-..../..-./-
-.../----./....-/....-/--.../.----/-..../-.../--.../-.../-..../..---
/...../.----/-..../-..../...--/....-/...--/-----/-..../...../--.../-..
很明显的摩斯密码，解密 744B735F6D6F7944716B7B6251663430657D
接着 base16 解密 tKs_moyDqk{bQf40e}
贴个解密网站 http://www.zjslove.com/3.decode/

tsmyq{Q4eK_oDkbf0}

接着栅栏解密，2 字一栏 tsmyq{Q4eK_oDkbf0}
接着凯撒解密，注意大小写，比如这个网站解密出来的忽略大小写了，还原就好，
flag:hagme{E4sY_cRypt0}

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:poium{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:hgame{e4sy_crypt0}
第14次解密:gfzld{d4rx_bqxos0}
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_ujqhl0}
第22次解密:yxrdv{v4jp_tipgk0}
第23次解密:xwqcu{u4io_shofj0}
第24次解密:wvpbt{t4hn_rgnei0}
第25次解密:vuoas{s4gm_qfmdh0}
第26次解密:utnzr{r4fl_pelcg0}

0x02 base 全家

正如题目名字一样，真的用到了 base 全家 orz,直接利用 python 的 base64 模块解密,第一次 base64 解密太长了,我直接在网站上解了,然后接下来,利用 python base64 模块直接解，至于到底用哪一个，base64 还是 base32,base16 使用最强的肉眼观察法，半手动进行解密，解的我自闭了 orz，我太菜了 orz，当然多亏

了这个，我现在能在 1S 内分辨出是 base64 还是 base32 还是 base16
最后解出来这个：
base58: 2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8Bn
然后找个网站 base58 解密就好
Flag:hgame{40ca78cde14458da697066eb4cc7daf6}