unserialize3

解题思路：

- 首先看到题目，知道是要做反序列化的相关工作，什么是反序列化？

  [这篇帖子](https://xz.aliyun.com/t/3674)写得很不错，详细介绍了 **“什么是序列化和反序列化”**，以及 **“常见的反序列化漏洞利用方法”**。

  简单来说：序列化就是让 **一个对象** 变成 **一个字符串*s*，

  反序列化就是让 **一个字符串** 变成 **一个对象** 。

- 点开链接，页面显示代码如下：

  ```php
  class xctf{
  public $flag = '111';
  public function __wakeup(){
  exit('bad requests');
      }
  }
  ?code=
  ```

- 一行一行阅读代码：

  首先，定义一个“xctf”类，这个类有两个成员属性，分别是 `flag` 和 `__wakeup` ，接下来是 `__wakeup` 的实现，再就是最后一行变量 `code` 没有写完，估计就是想让我们传参进去完成它了。

  这里的 `__wakeup` 函数是一个所谓的“魔术方法”，它在反序列化时会先于其他函数被调用，执行它的语句。

  这就意味着，如果我们直接构造一个字符串，那么这个 `__wakeup_` 会先一步执行，并直接 exit，这可不是我们想要的结果。

- 读完代码，准备解题，怎么样才能绕过这个 `__wakeup` 函数呢？查阅资料后发现，有一个漏洞（CVE-2016-7124）可以通过 **使序列化字符串中表示对象属性个数的值大于真实的属性个数，以此跳过__wakeup 的执行** 。

  这句话怎么理解呢？首先，序列化字符串的标准格式：

`O:<类名的长度>:"<类名>":<成员属性的个数>:{S:<成员属性名的长度>:"<成员属性名>";.....}`

这道题的话，如果要对 xctf 类进行正确的序列化，那么它的字符串应该是：

`O:4:"xctf":1:{S:4:"flag";S:3:"111";}`

- 其实这里有一个小问题，我看到资料上说反序列化的目标是一个对象，那么现在的 `xctf`类到底可否被看作一个对象呢？

- 再仔细看了一下，其实本题中序列化的目标本身就是一个对象，根据 php 官方文档显示：

[序列化一个对象将会保存对象的所有变量，但是不会保存对象的方法，只会保存类的名
字。](https://www.php.net/manual/zh/language.oop5.serialization.php)

- 这就意味着我们不必在意对象的名称，只需要知道它是哪个类的实例化即可，表现在这道题中就是 `O:4:"xctf"` .

- 拿到了正确的串，现在我们考虑跳过 `_wakeup` 的事情：

首先，我们知道 xctf 里只有一个属性 `flag` ，那么我们就把串里的那个 `1` 改成任意大于一的数，我随便改了个 5，结果如下：

`O:4:"xctf":5:{S:4:"flag";S:3:"111";}`

- 以 get 方式发出去，直接拼接 url：

`http://111.198.29.45:55168/?code=O:4:%22xctf%22:5:{S:4:%22flag%22;S:3:%22111%22;}`

- 服务器返回 flag 值，完成。