#### Thinkphp5.0.10-3.2.3缓存函数设计缺陷可导致getshell

## 0x00 框架运行环境

ThinkPHP是一个免费开源的，快速、简单的面向对象的轻量级PHP开发框架，是为了敏捷WEB应用开发和简化企业应用开发而诞生的。ThinkPHP从诞生以来一直秉承简洁实用的设计原则，在保持出色的性能和至简的代码的同时，也注重易用性。

Thinkphp在使用缓存的时候是将数据 序列化 然后存进一个php文件中这就导致我们了我们在一些情况下可以直接getshell

0x01 漏洞利用

该漏洞形成最关键的一点是需要使用框架时，有使用缓存，才能利用这个漏洞

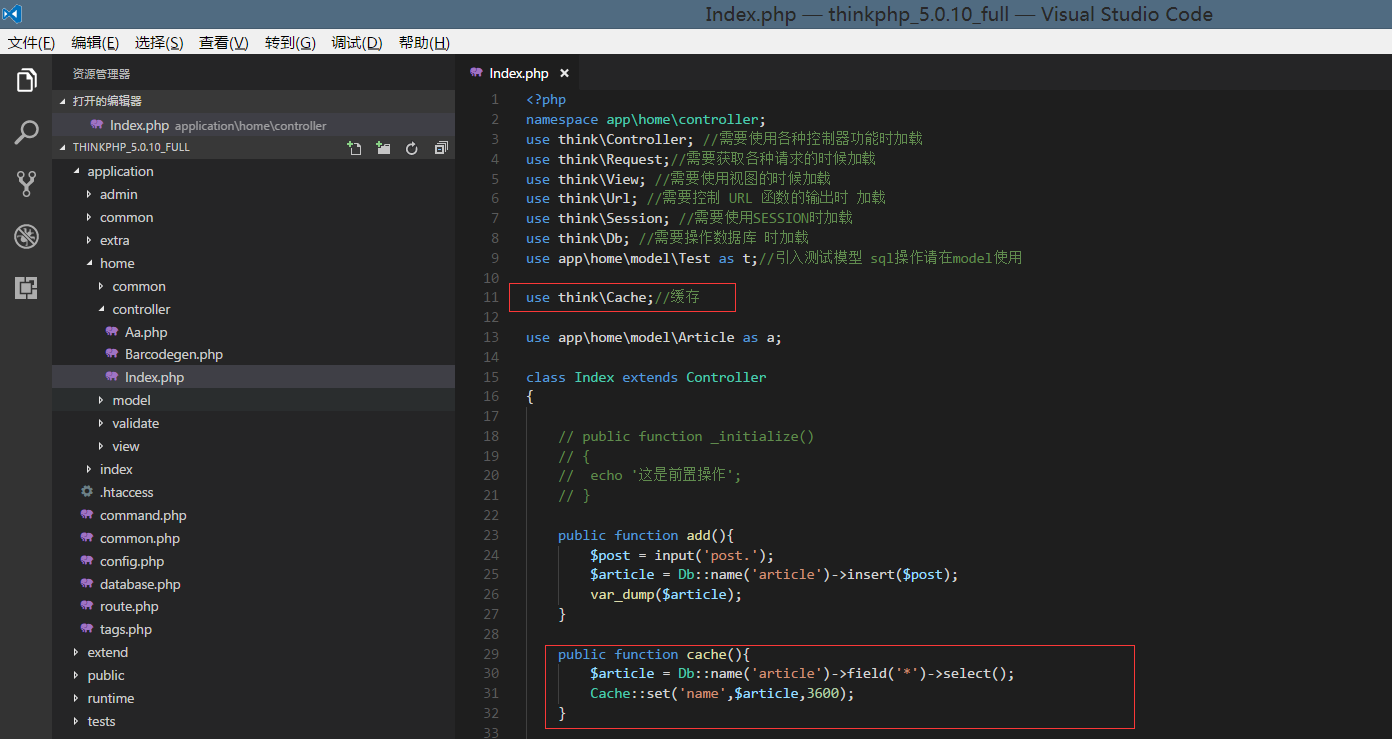


我们这里使用缓存查看官网对这个缓存类的说明以及利用方法

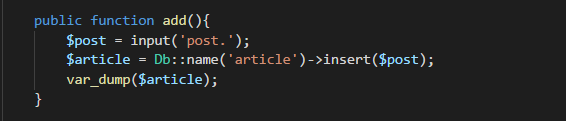
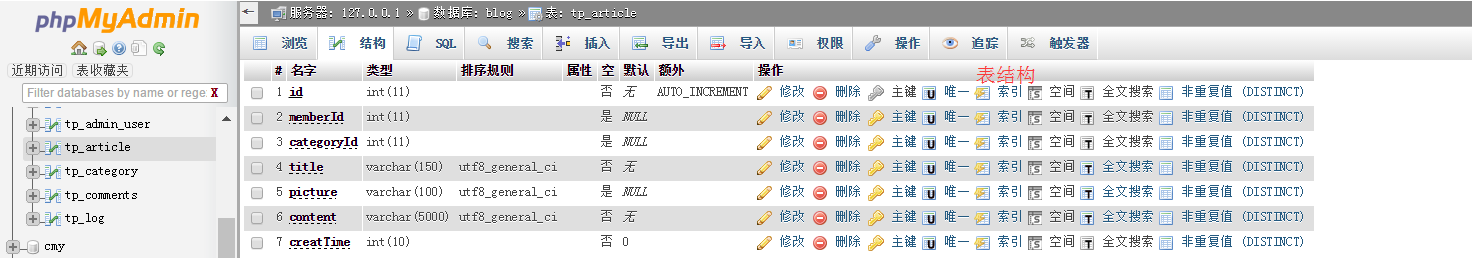
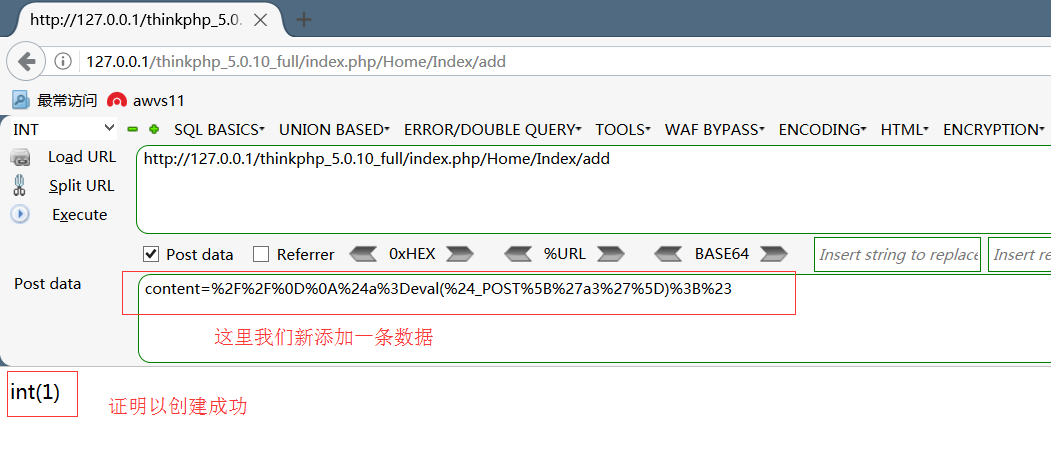




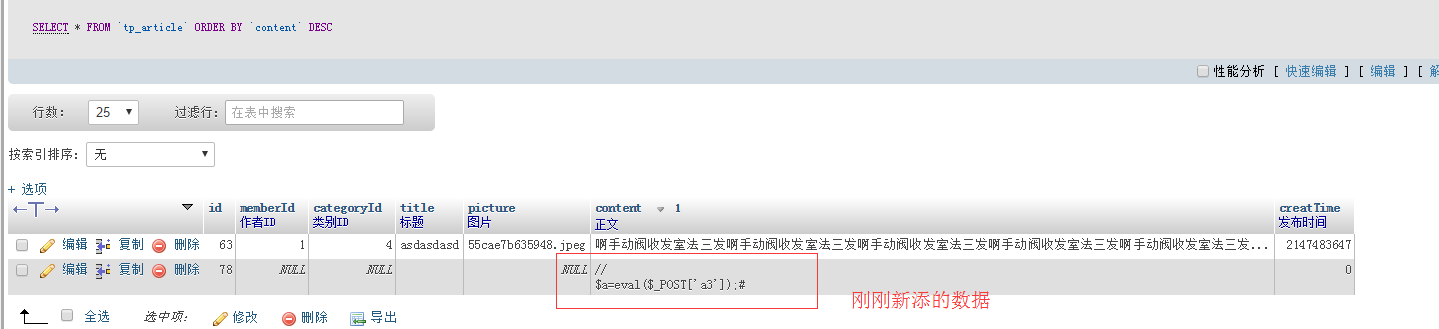
本地按照官方给的文档安装成功后，根据官网给的缓存使用方法，新建一个方法，我们都清楚缓存一般是为了减少数据库的开销为设置的，所以缓存的数据一般也是从数据库获取到的 为了模拟线上，我们这里先查数据库数据在写入缓存。



这里我们写了一个add添加数据的方法

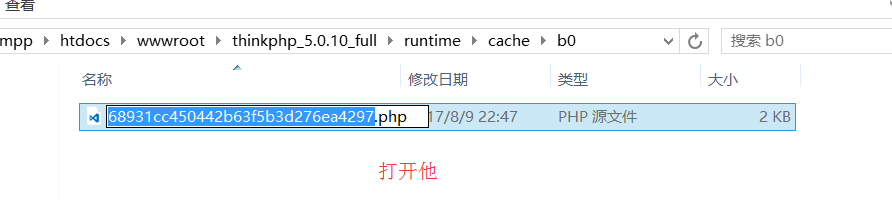
  

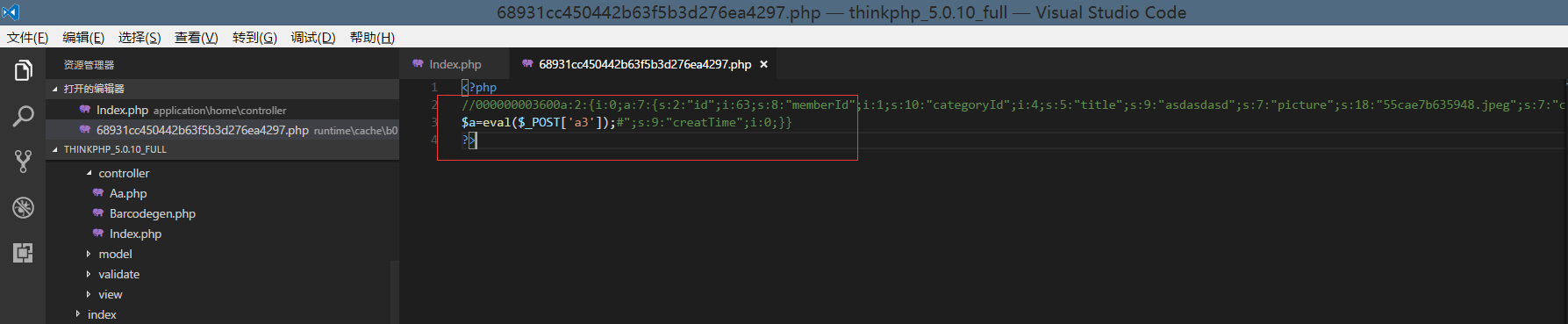
%2F%2F%0D%0A = //+回车



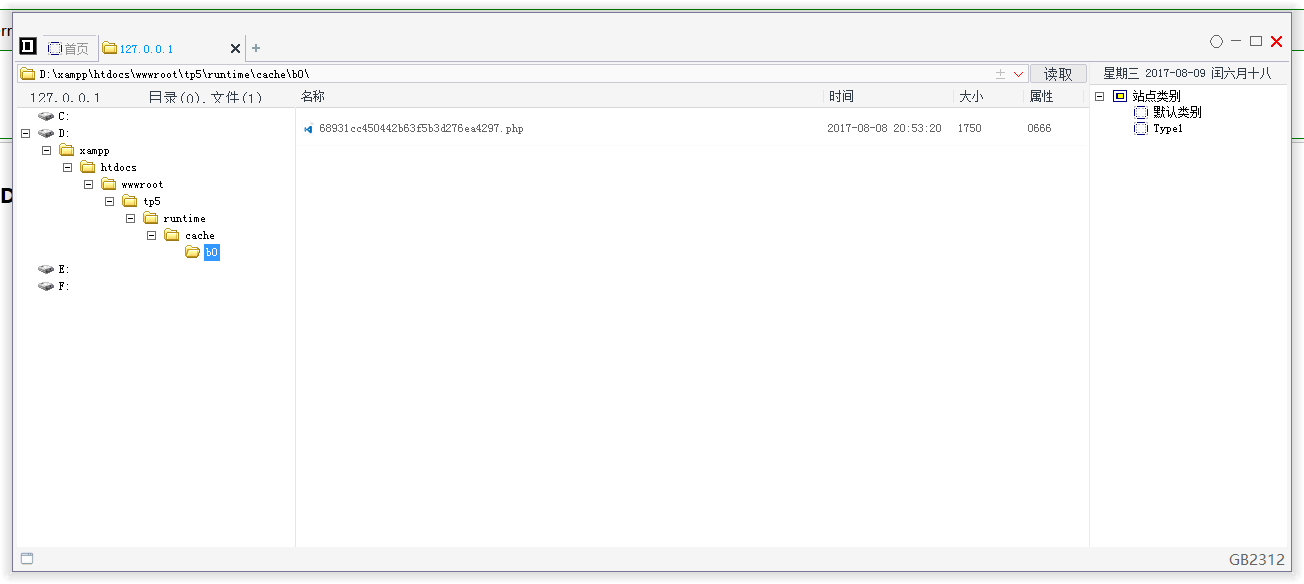


执行完以后查看方法缓存目录



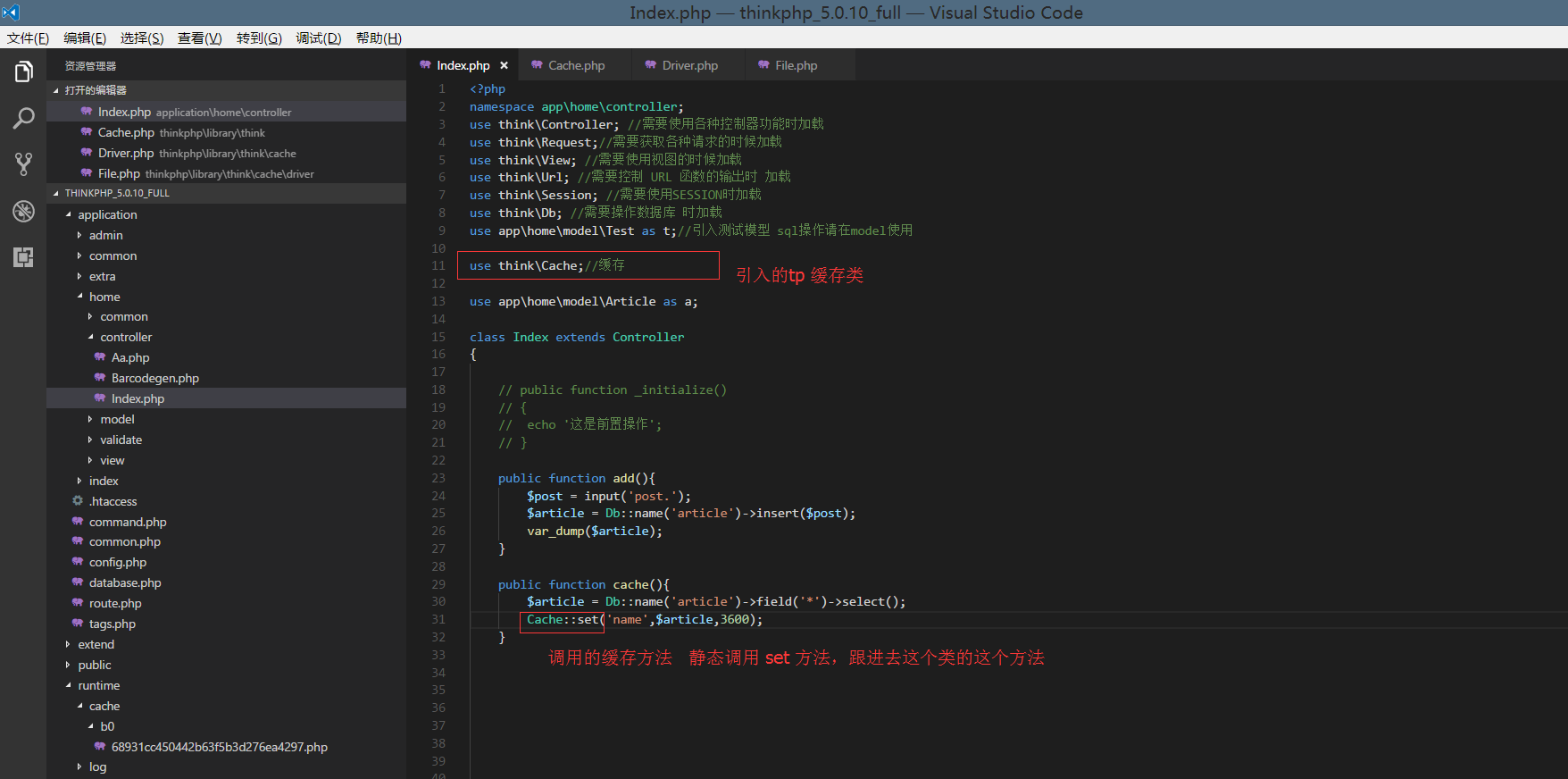


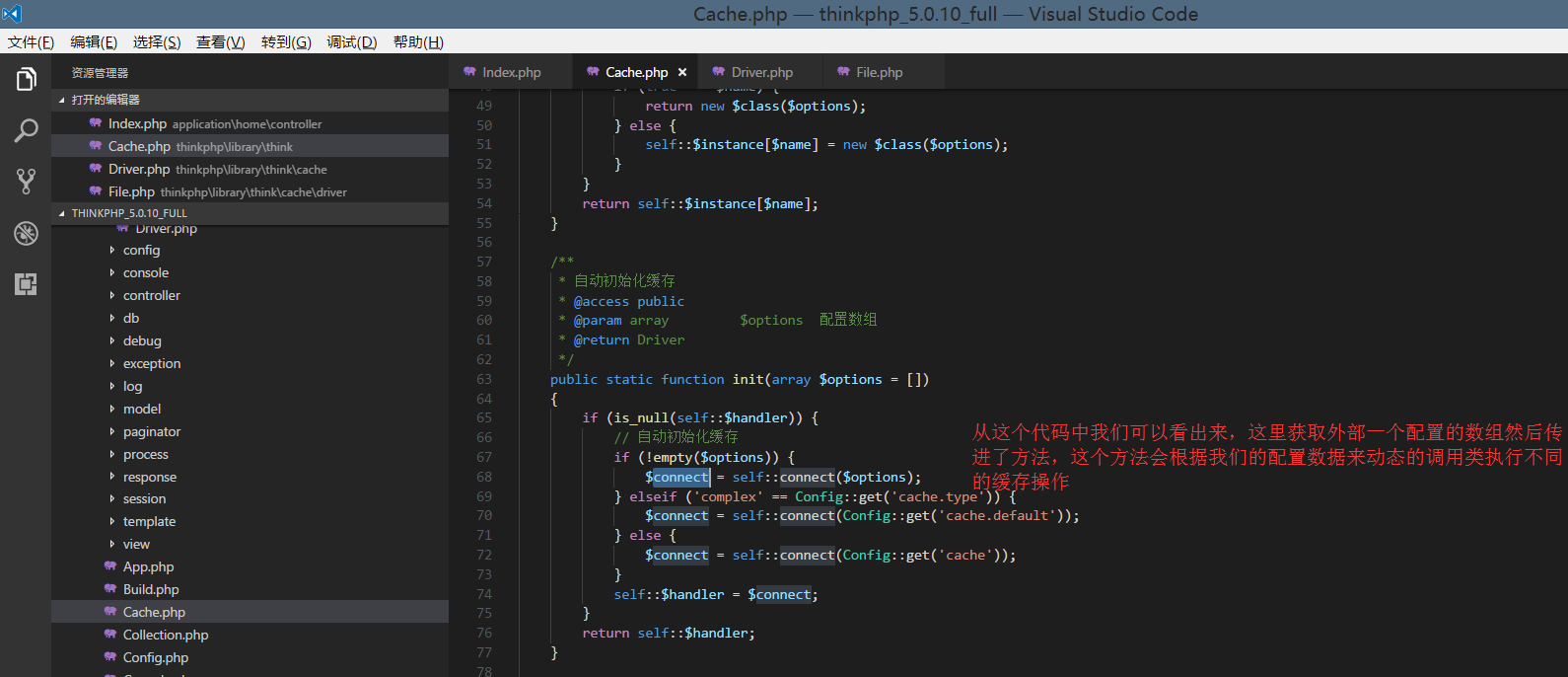
这里需要特别说的一点是TP的缓存名字是不变的，所以我们在审计的时候不用怕缓存文件名猜不到的情况。

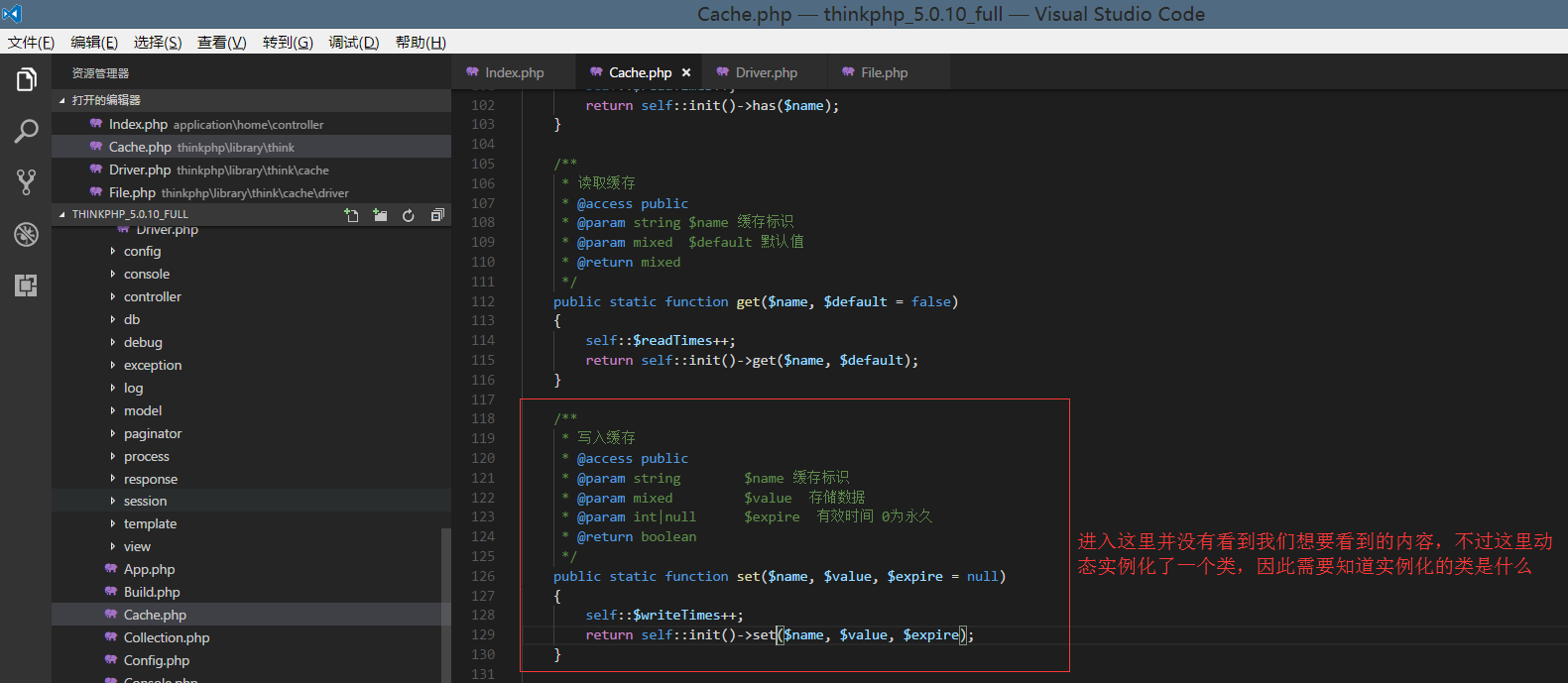


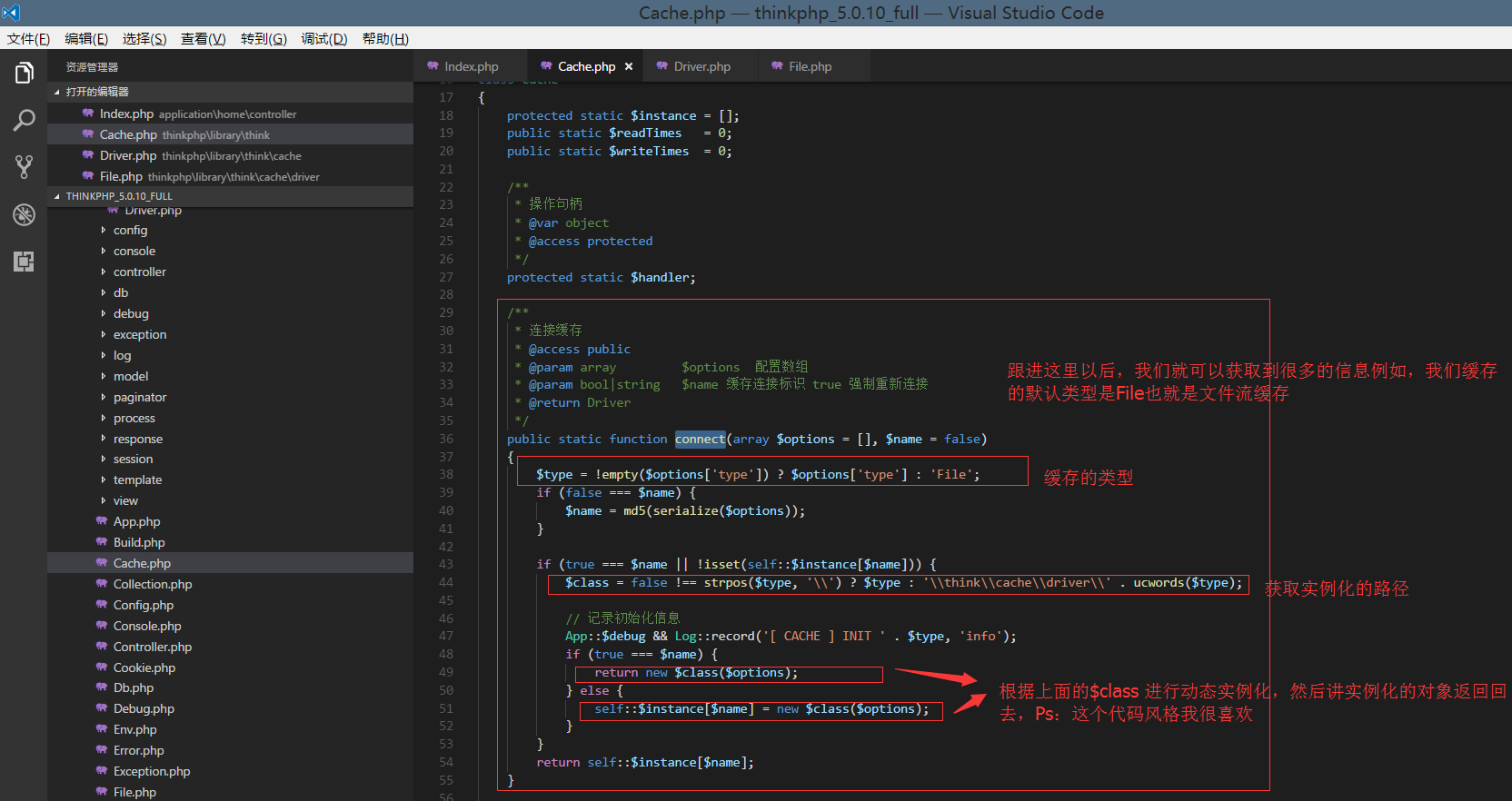
0x02漏洞分析

上面我们展示了漏洞利用方法，这里我们对这个漏洞进行分析



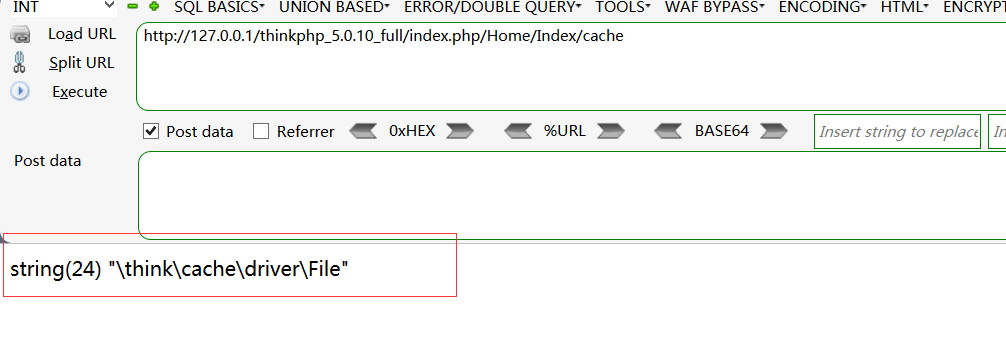




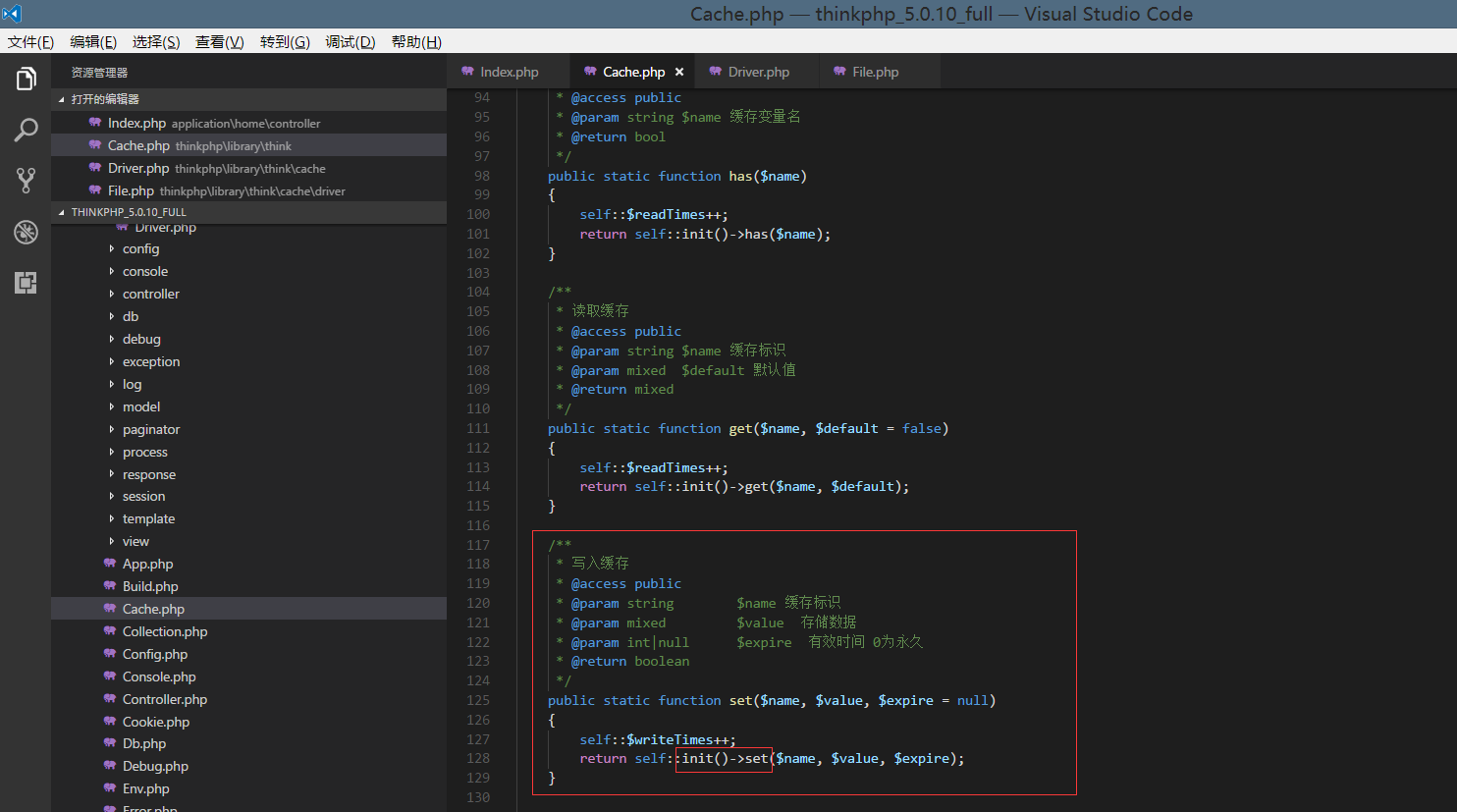


为了证明我们的逻辑是对的我们这里打印一下返回的数据

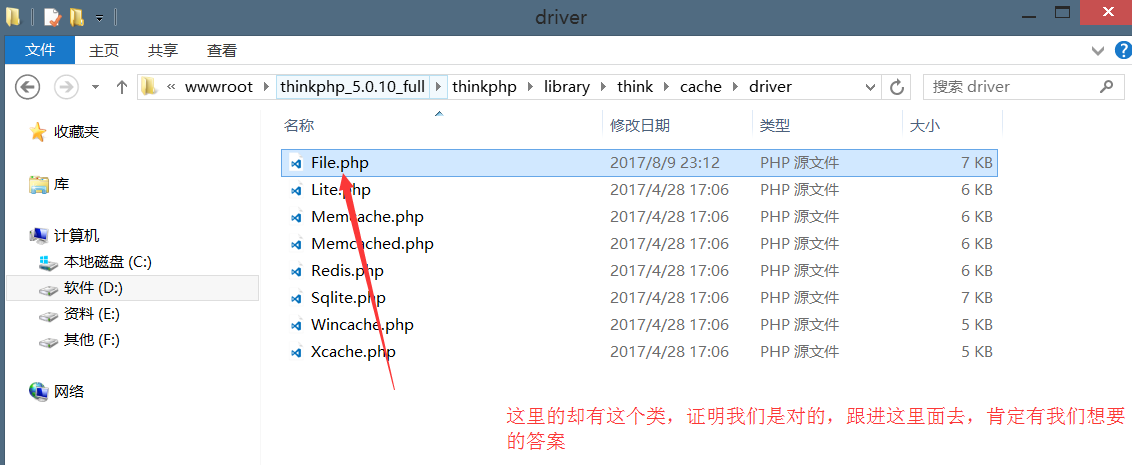


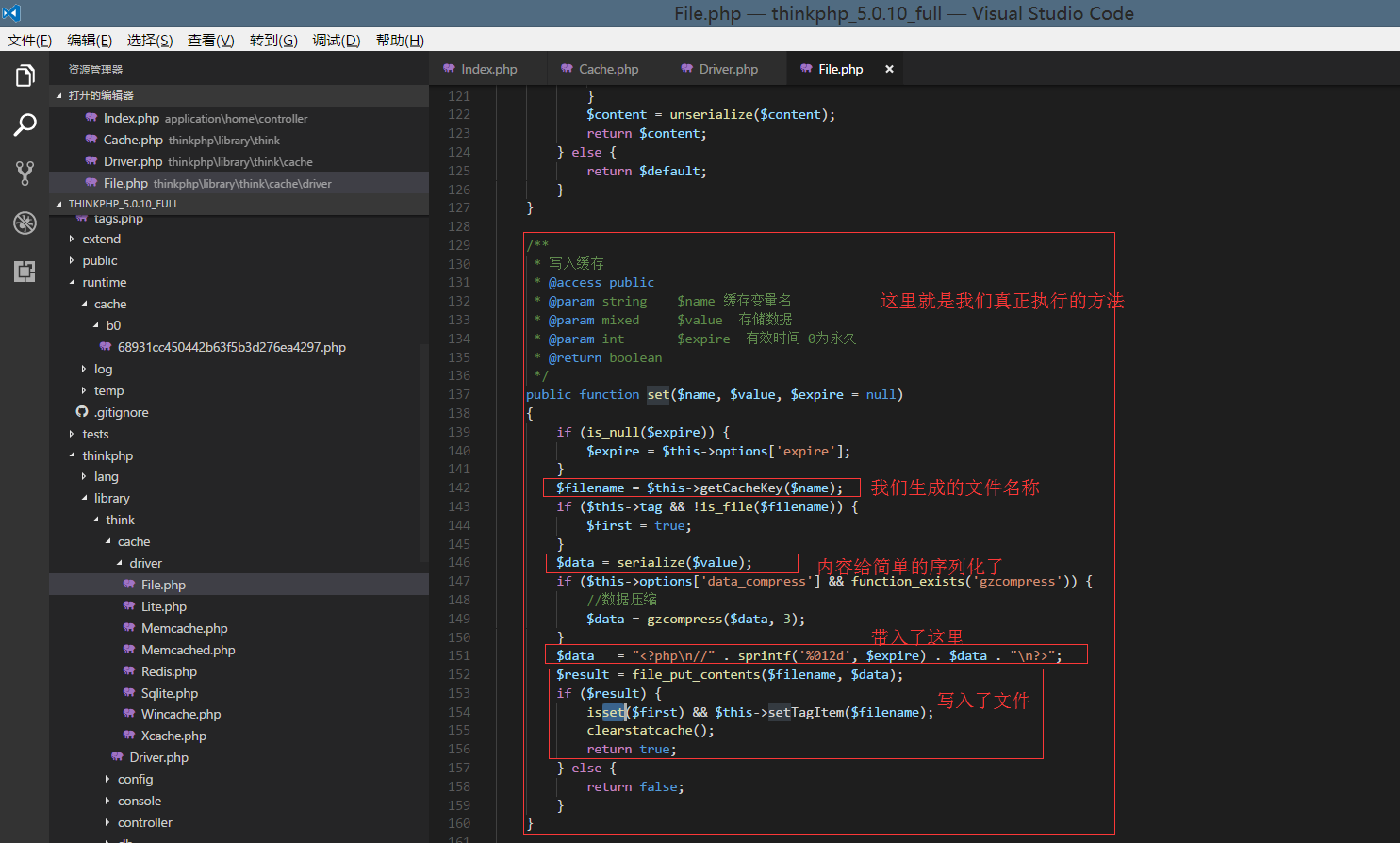


通过这个我们就可以知道了下面这个截图的意思

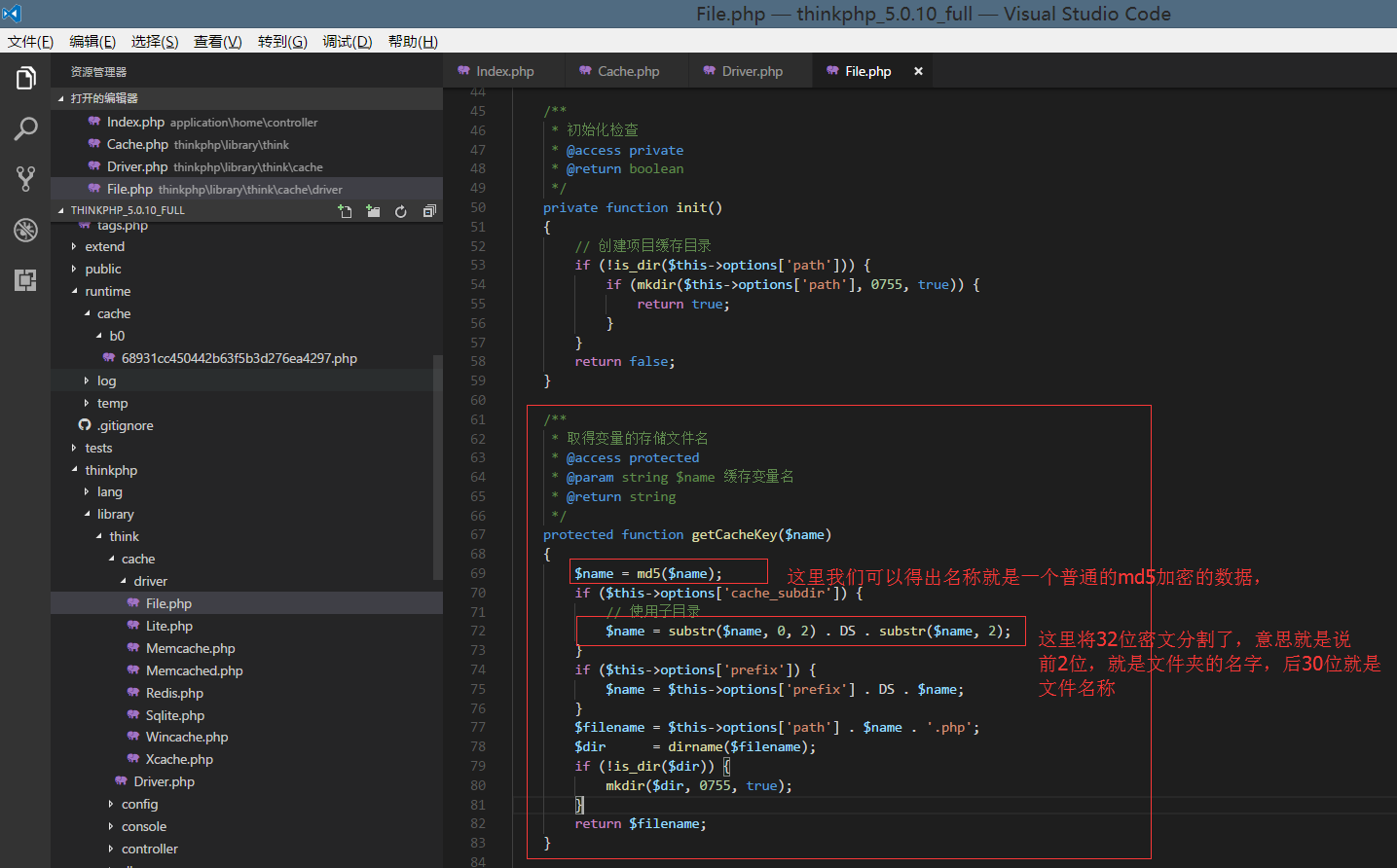


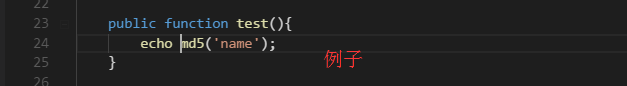
实例化 \thinkp\cache\driver\ 文件里面的 File类 并且调用set方法



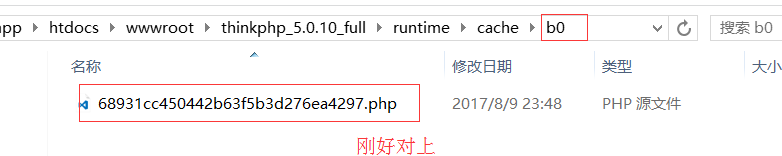


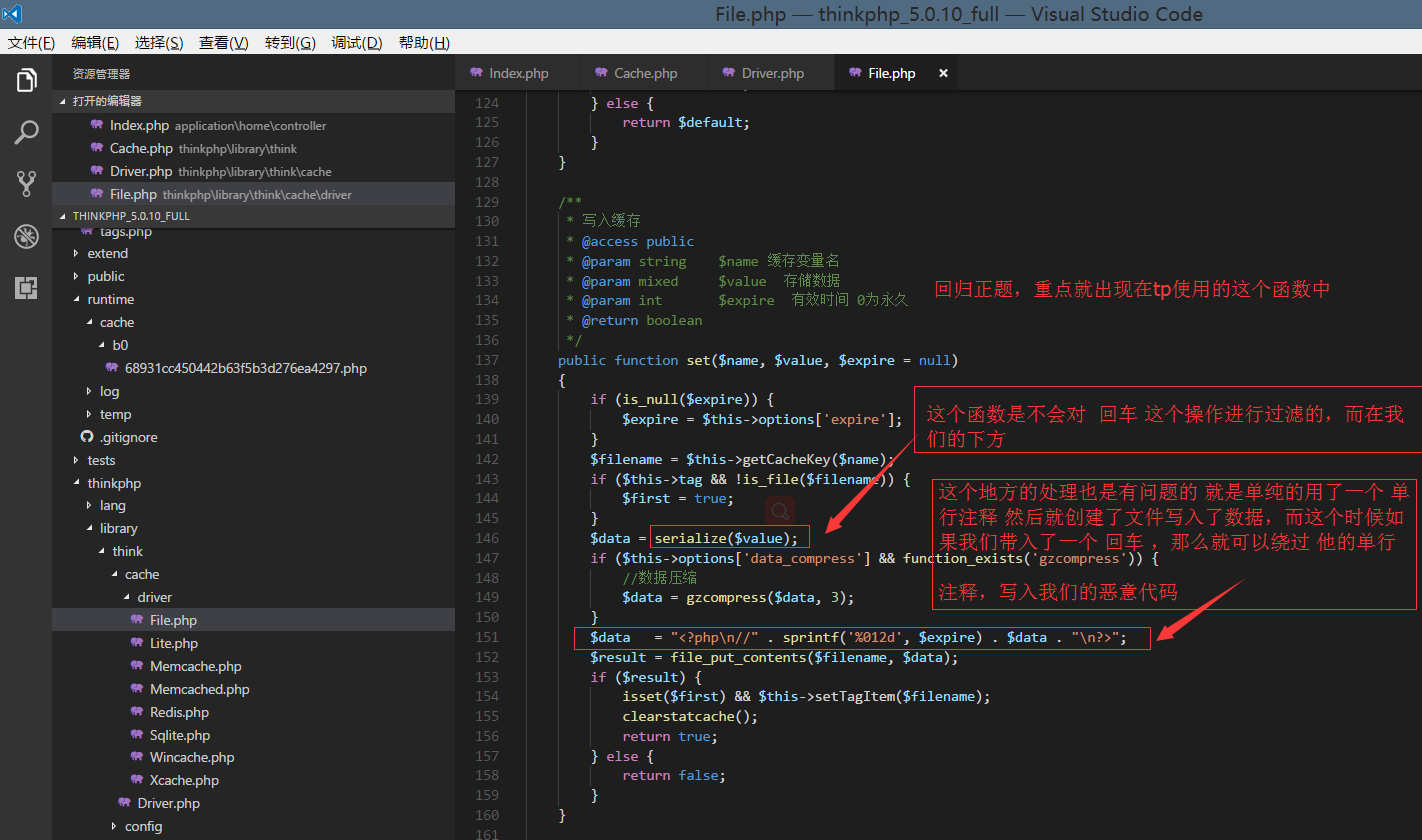
缓存文件名称的获取方法







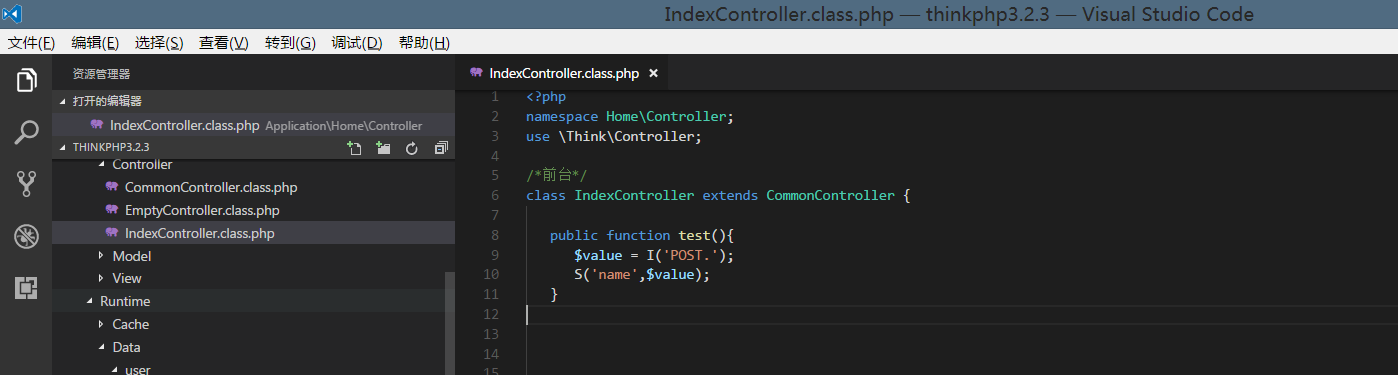




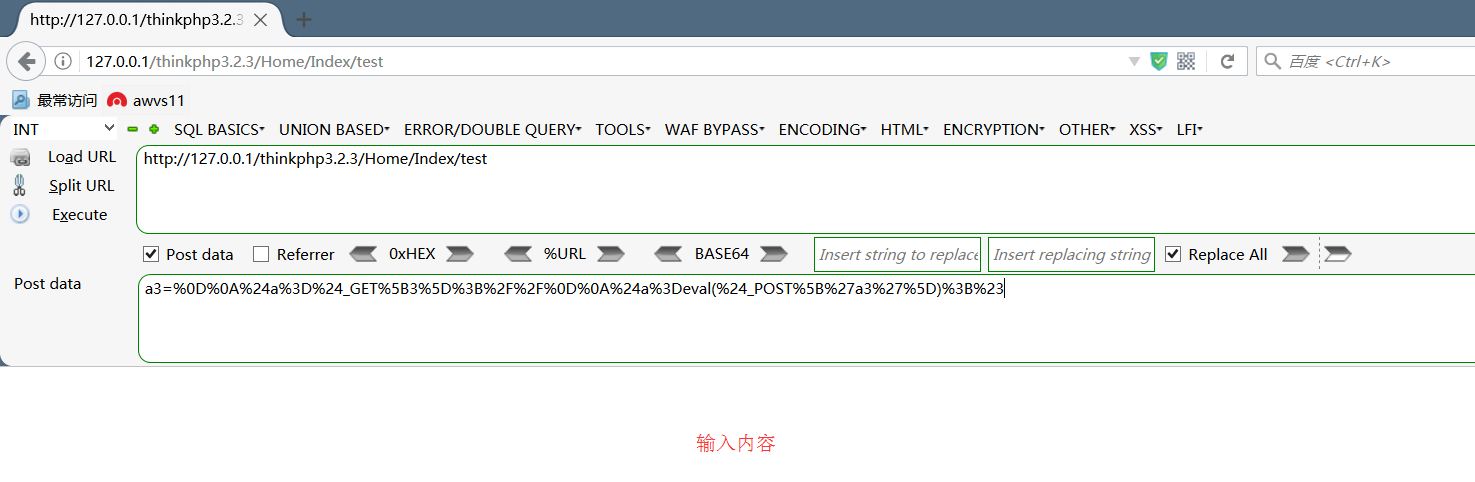
# Thinkphp3.2 缓存函数设计缺陷

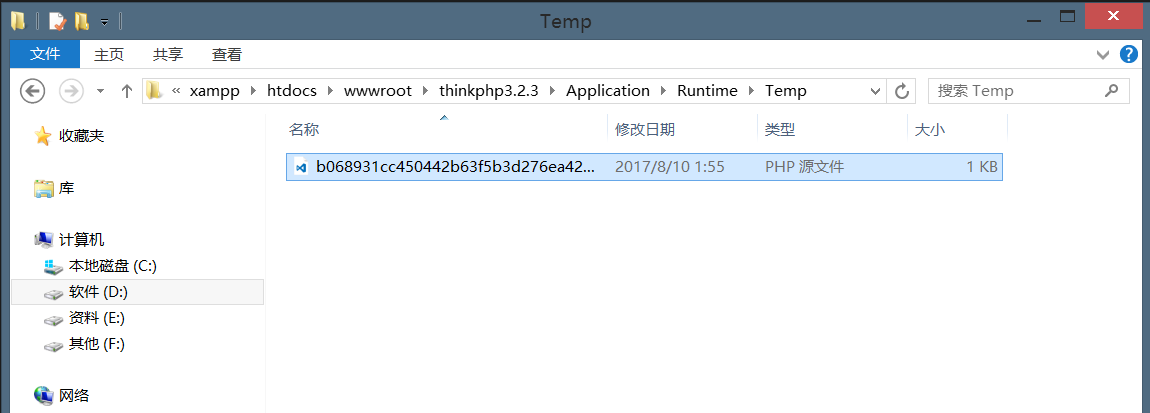
这个感觉没什么可以说的，和上面的原理是一样的，我们只演示攻击的方法

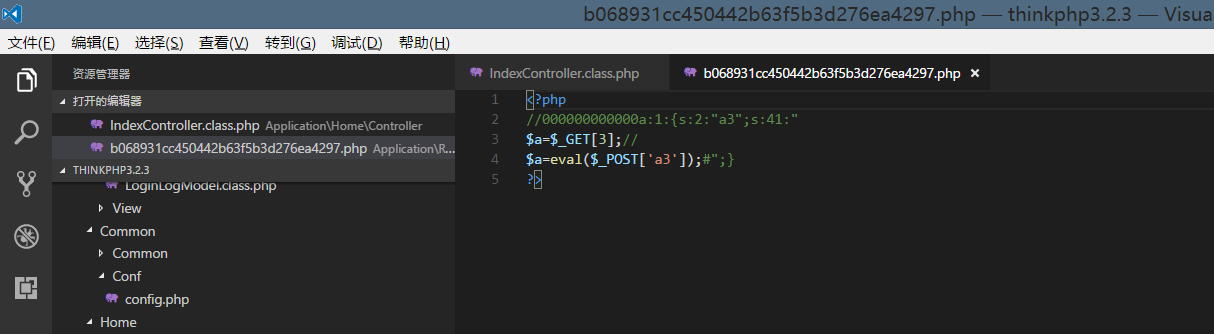


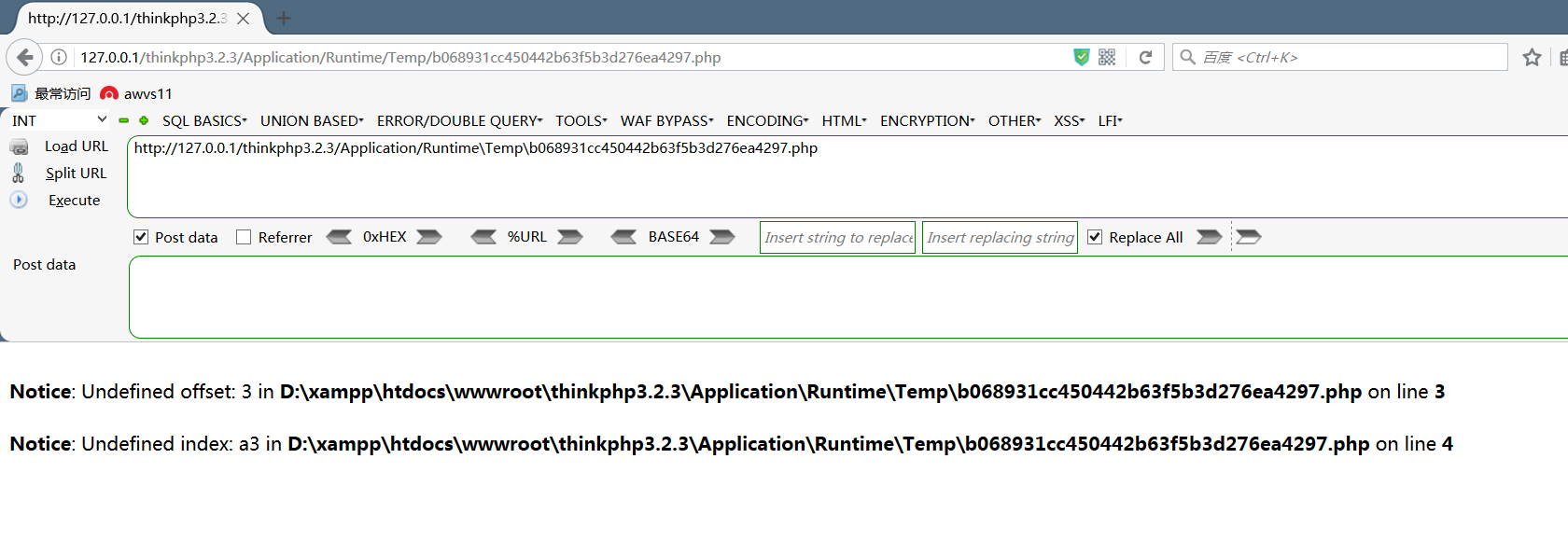


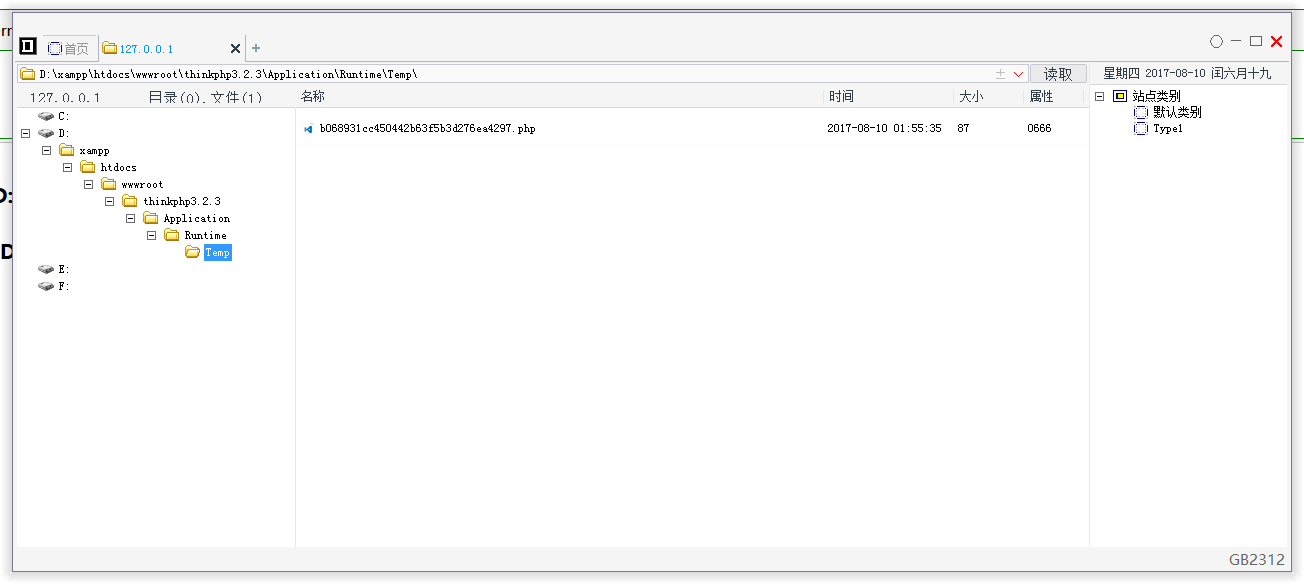












通过上面的过程与分析我们可以清楚了解造成这个漏洞的主要原因就是换行与回车导致绕过了注释。那么我们修复的方法就很简单了只要删除这些即可

修复方法：

1. 打开文件：thinkphp\library\think\cache\driver\File.php
2. 找到：public function set($name, $value, $expire = null) 方法
3. 添加：$data = str\_replace(PHP\_EOL, '', $data); 如下图
4. 