## ThinkPHP5.0.10框架filterExp函数过滤不严格导致sql注入

0x00 说说说说说说说说说说说明

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。

filterExp函数thinkphp5框架核心的安全过滤函数，他被配置与input函数一起使用， 他的前生是I函数，thinkphp5重写了数据库操作类方法，filterExp函数没有及时更正更新导致sql注入。

0x01漏洞利用

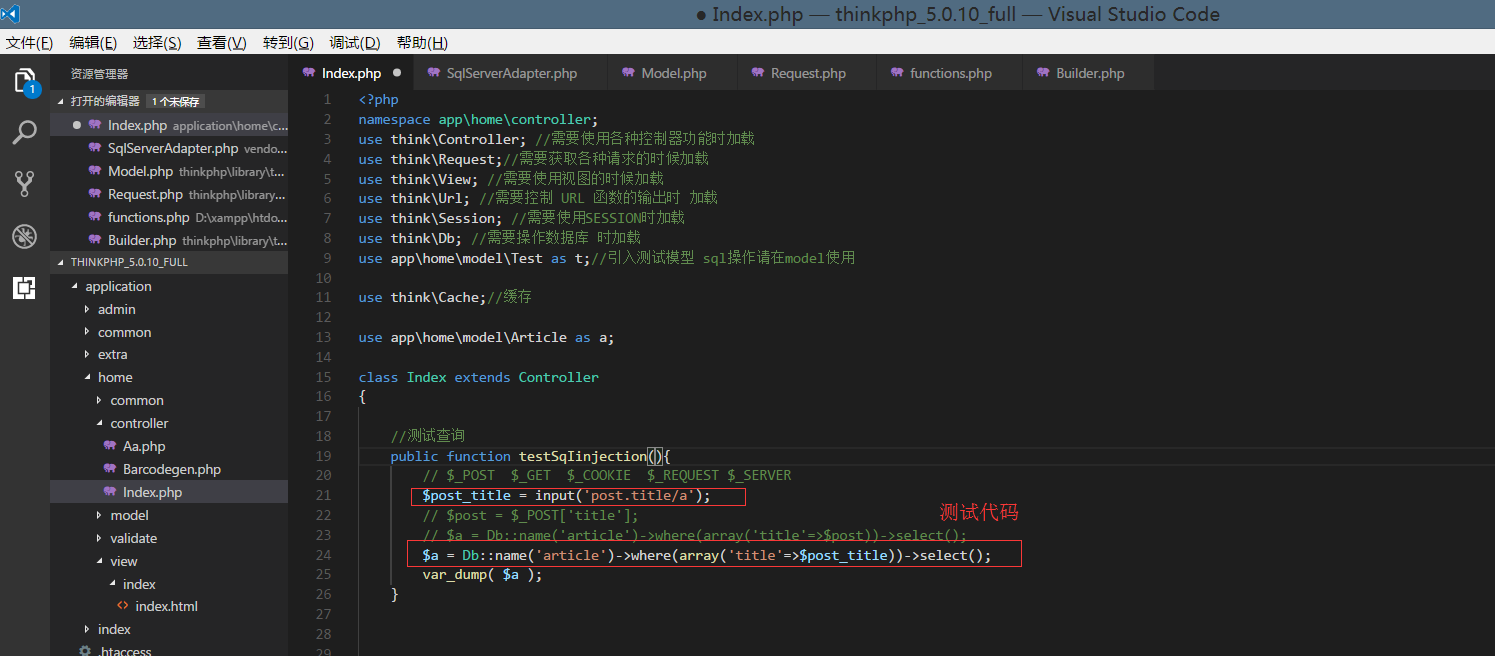


凌晨5点了，差不多猝死了，原理什么的就放在最后面了，直接来看如何利用把。QAQ

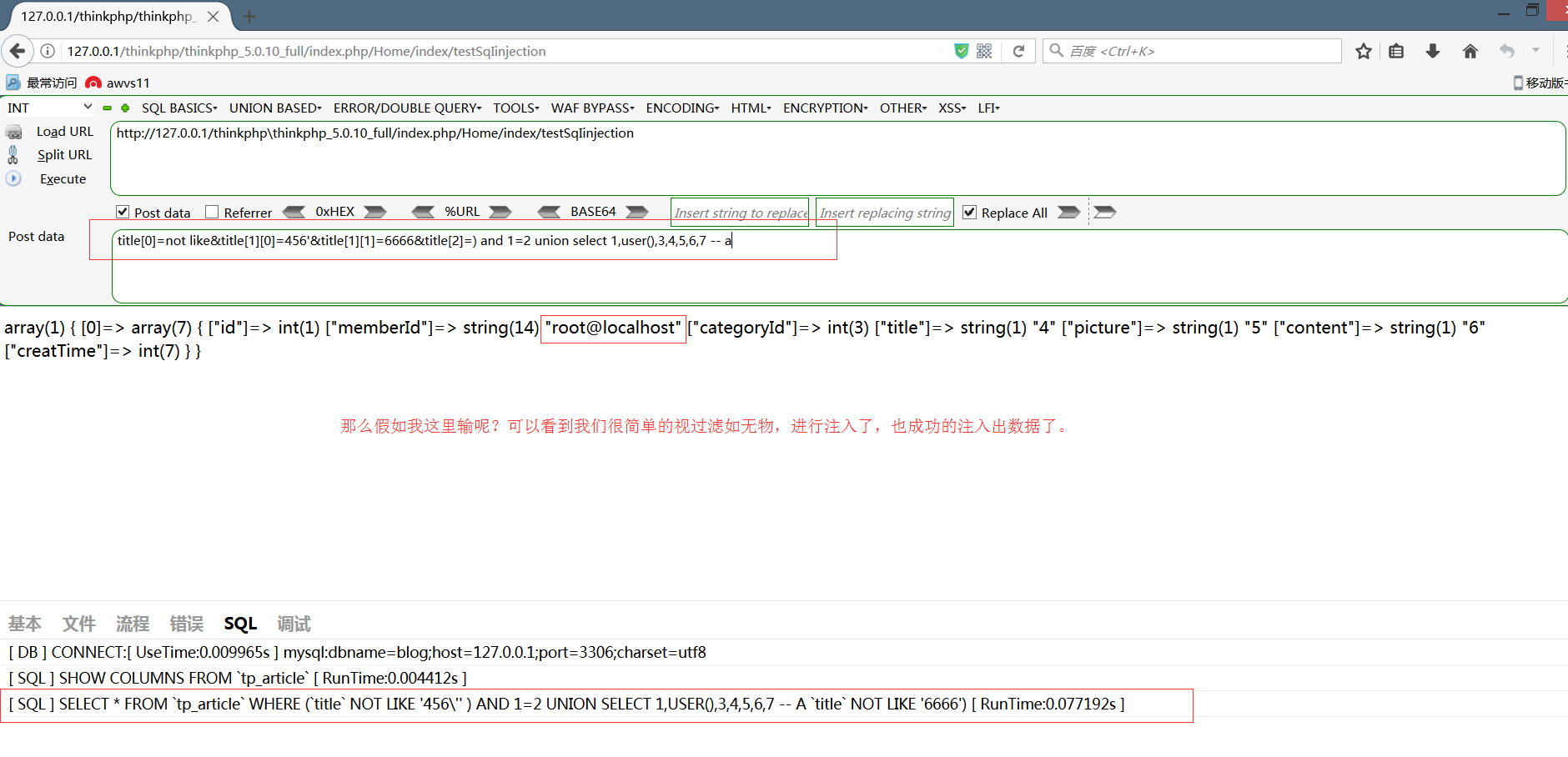
1，首先先看一波 input函数 了解这个东西何物先

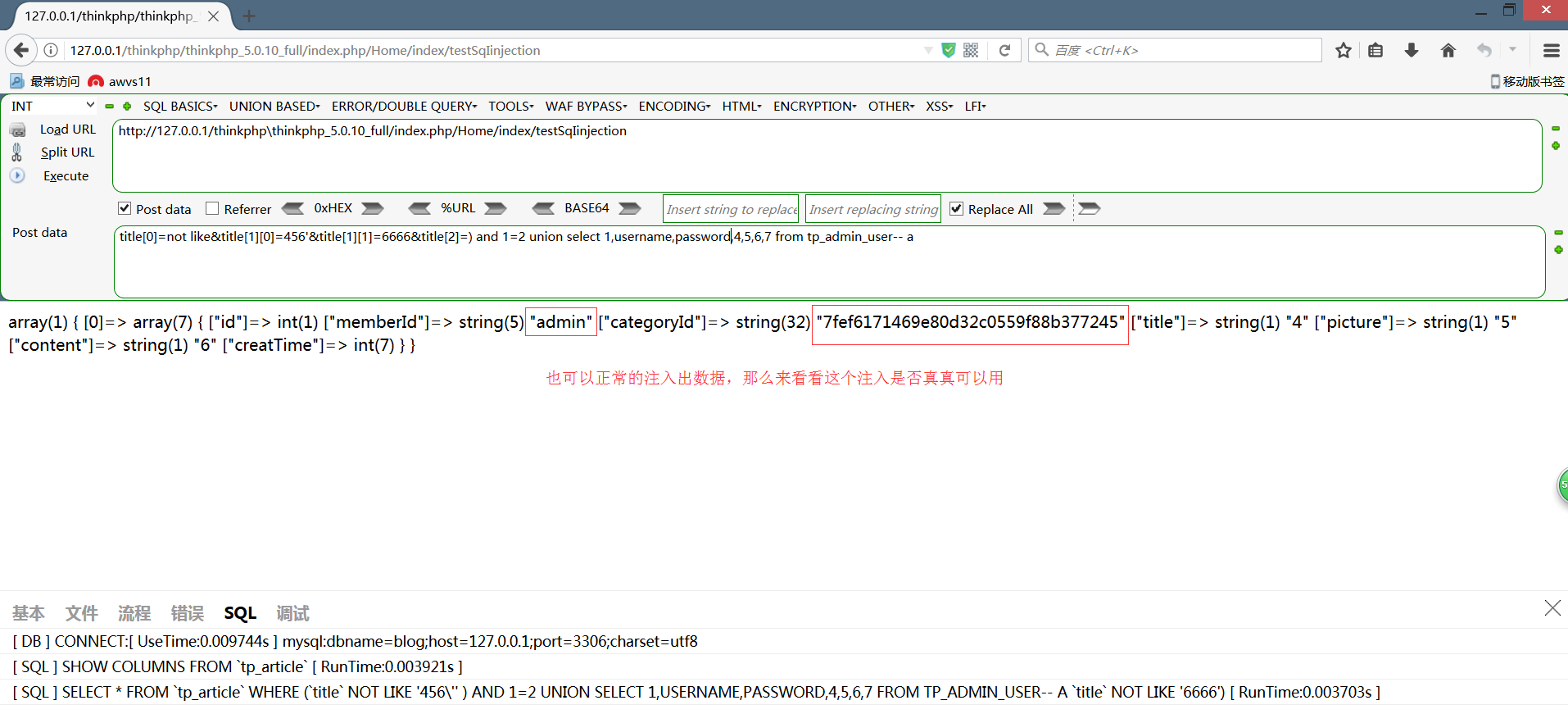


漏洞场景方法1：

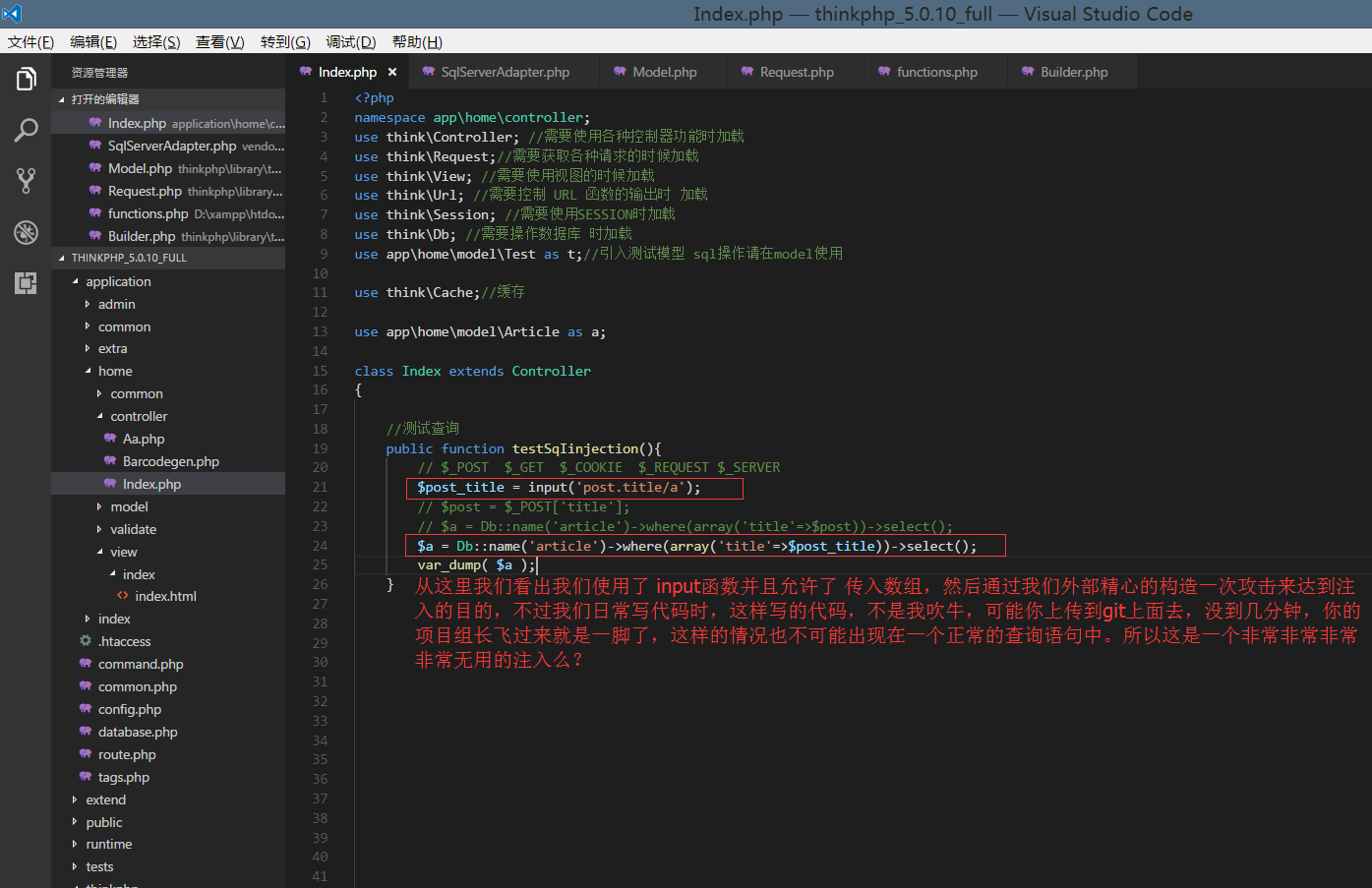


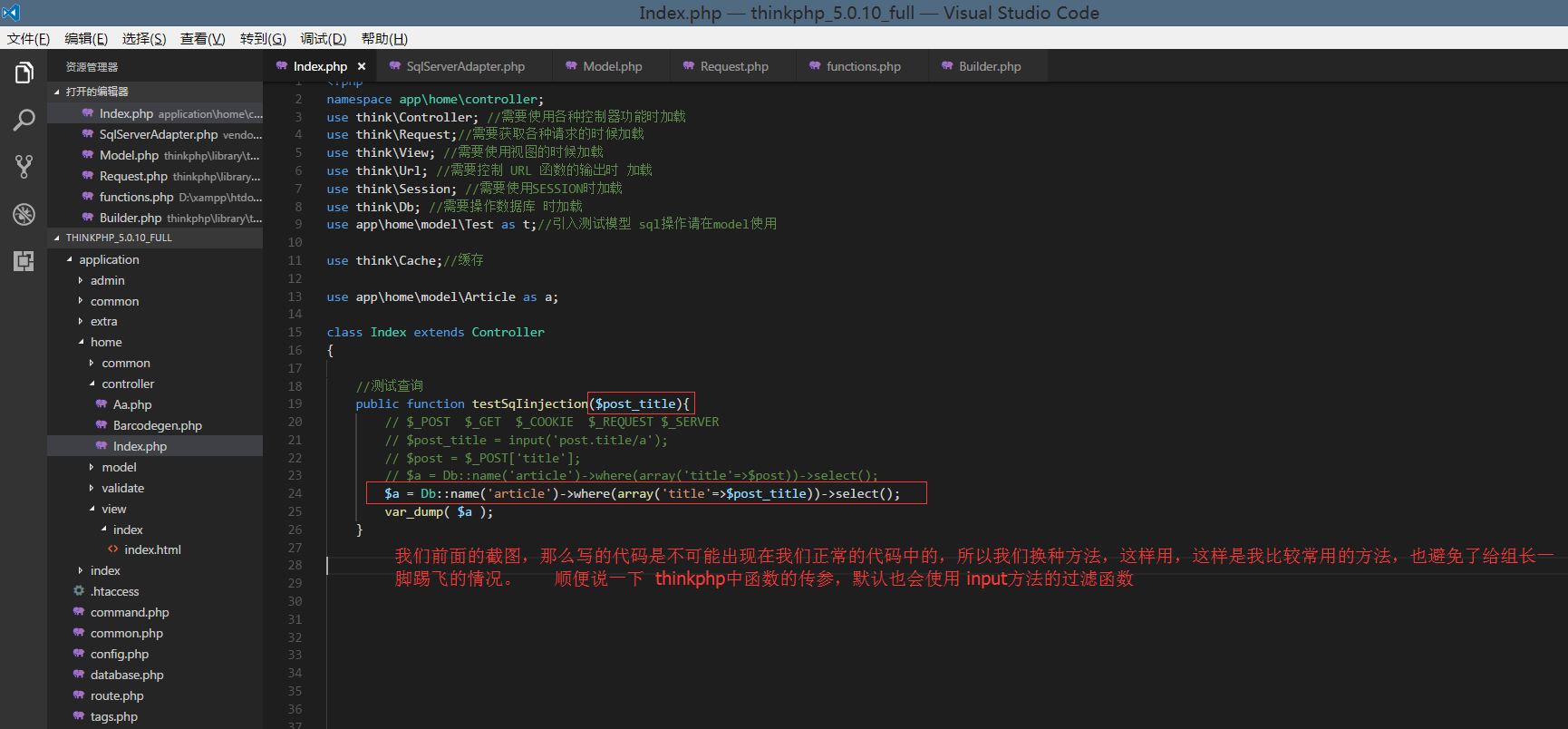


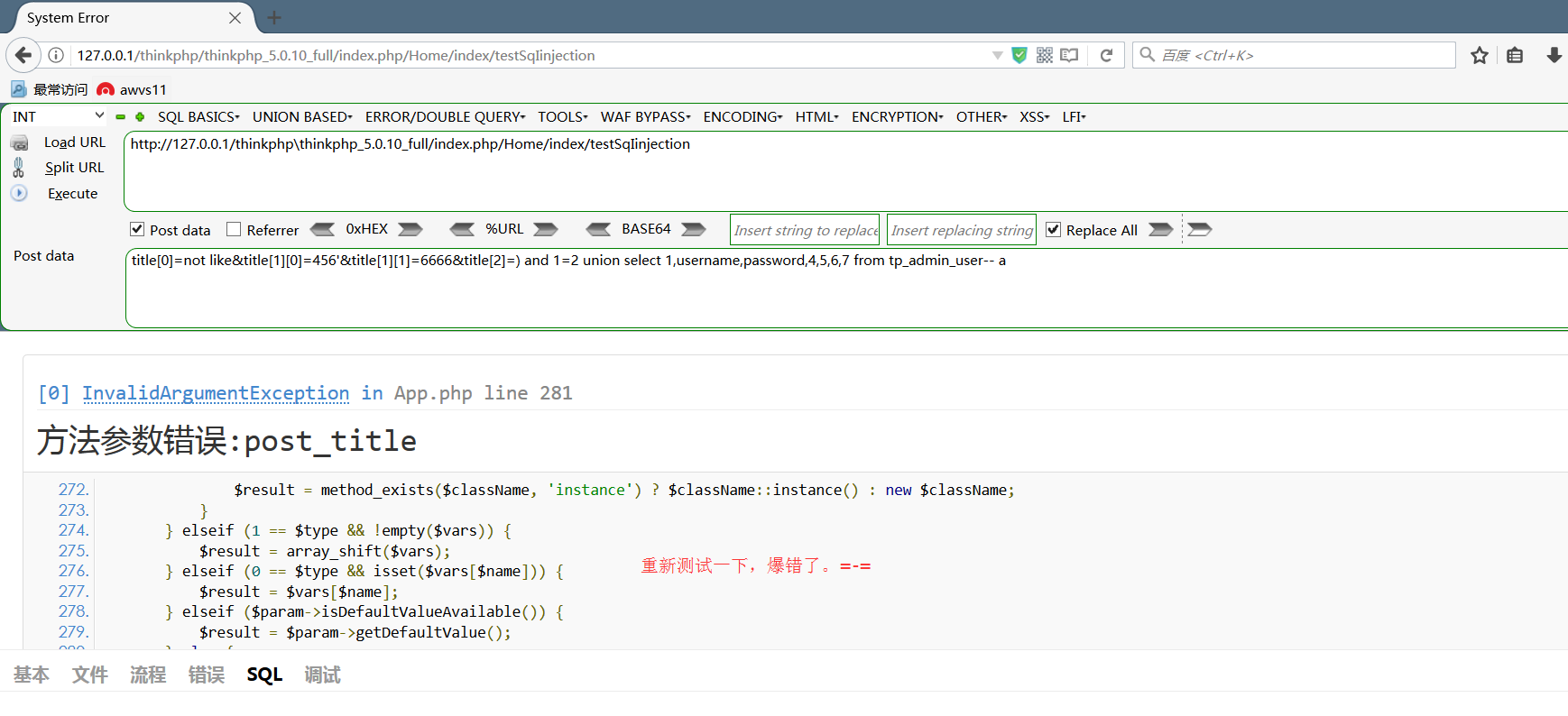


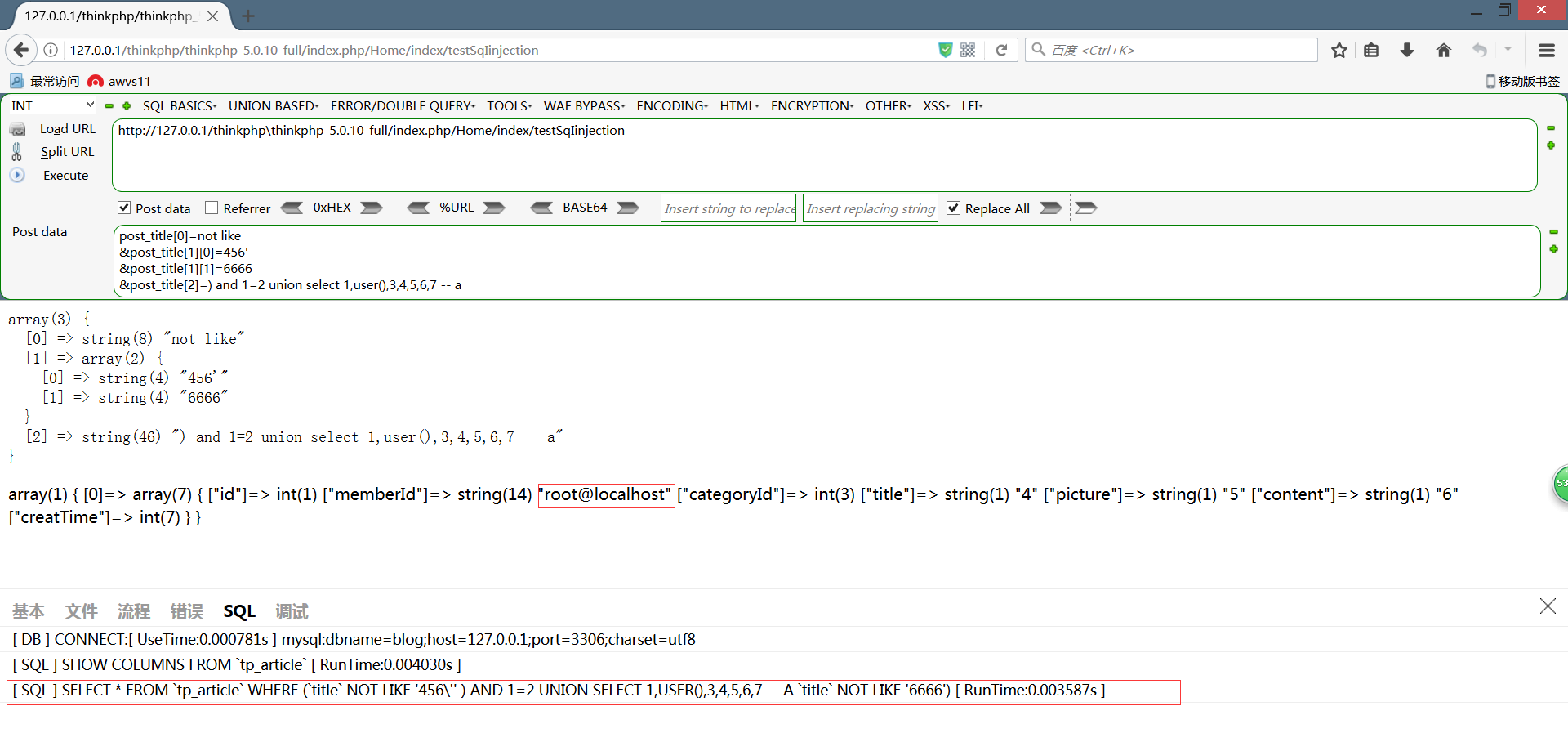


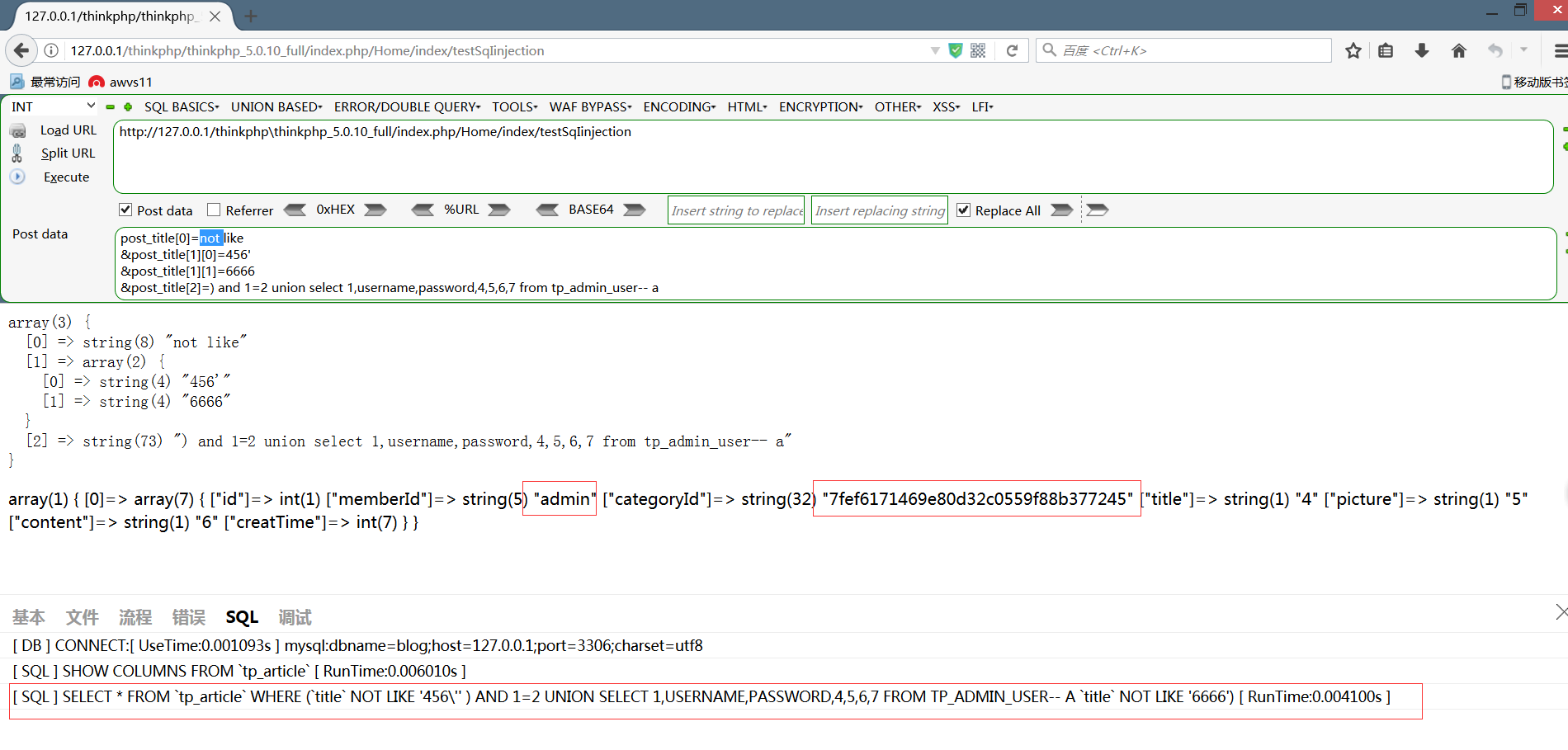
漏洞场景方法2：











0x02漏洞分析

上面我们展示了无聊的漏洞利用方法，这里我们对这个漏洞进行分析

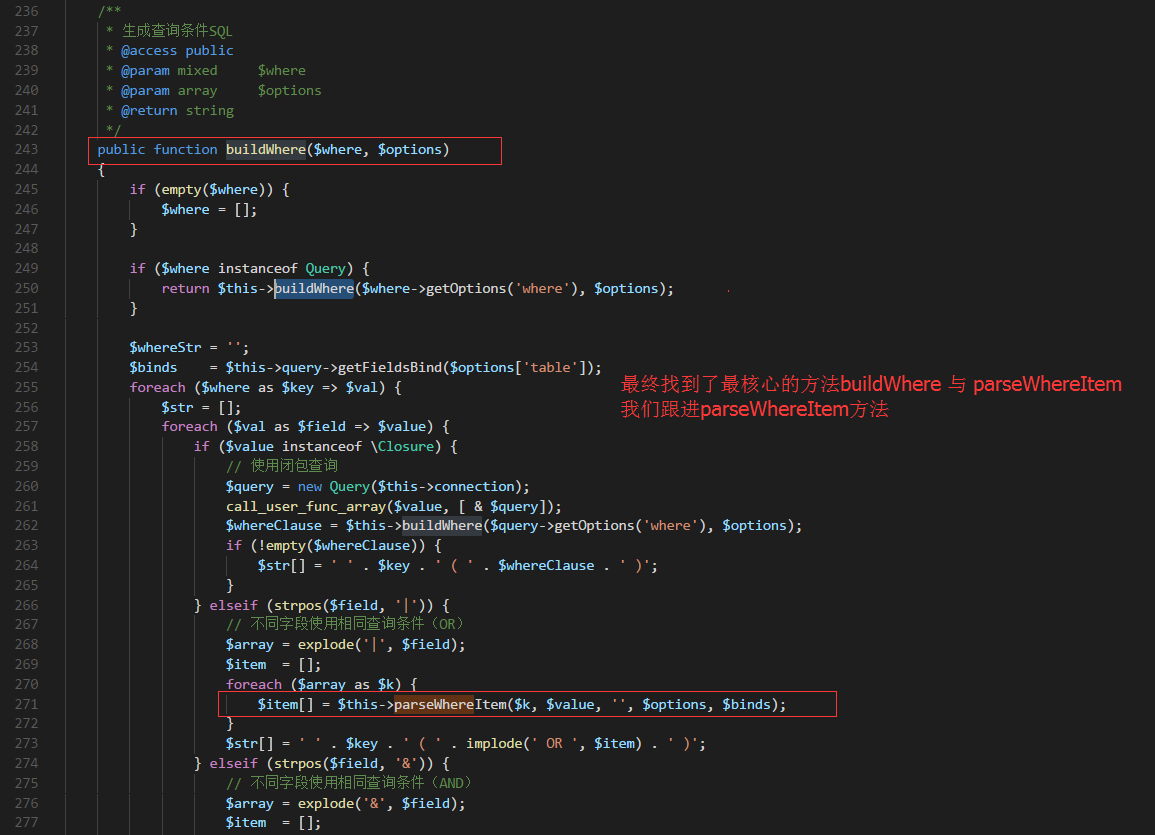
那就看一下 select方法是如何定义的

文件：thinkphp\library\think\db\Builder.php

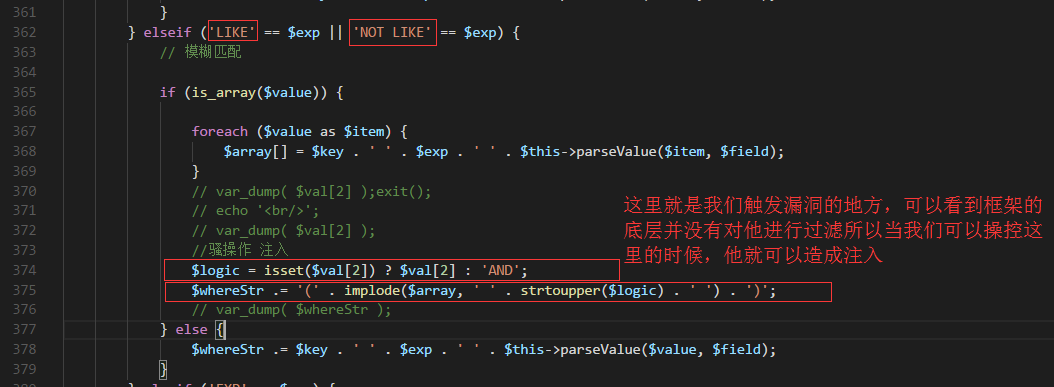
方法：select()







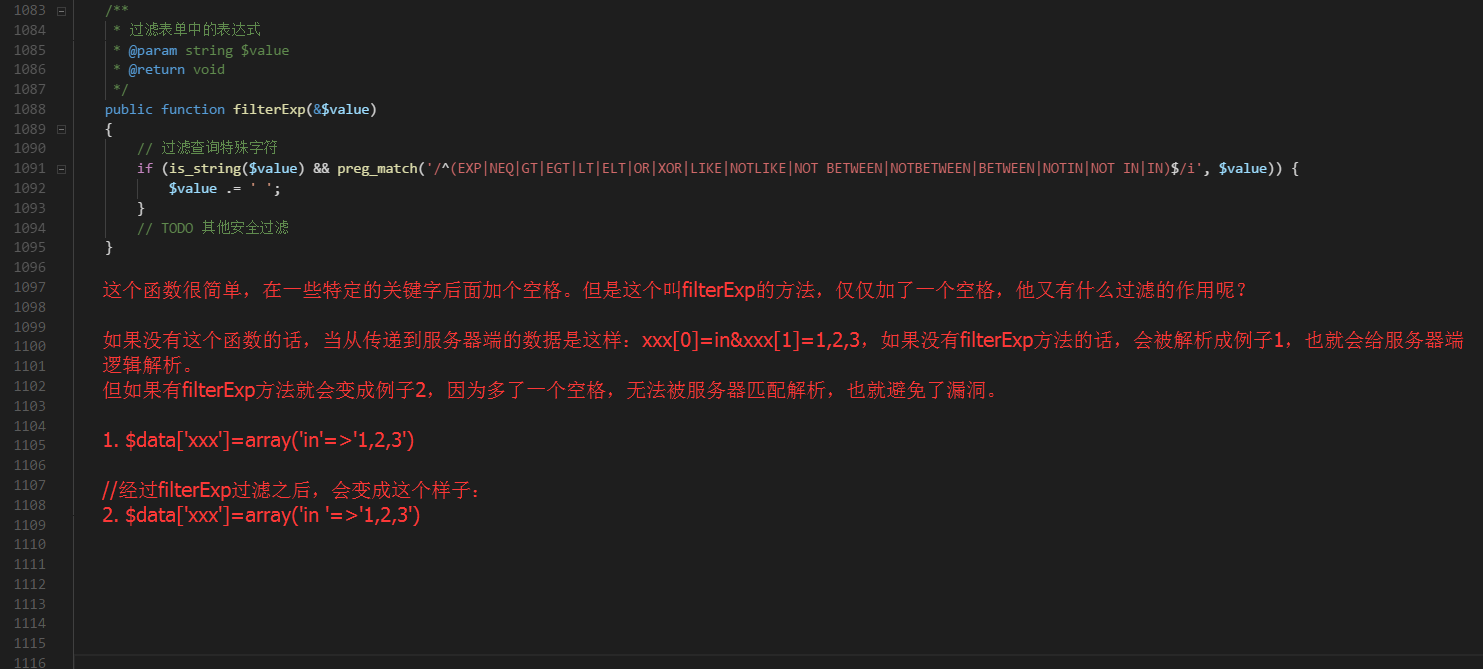
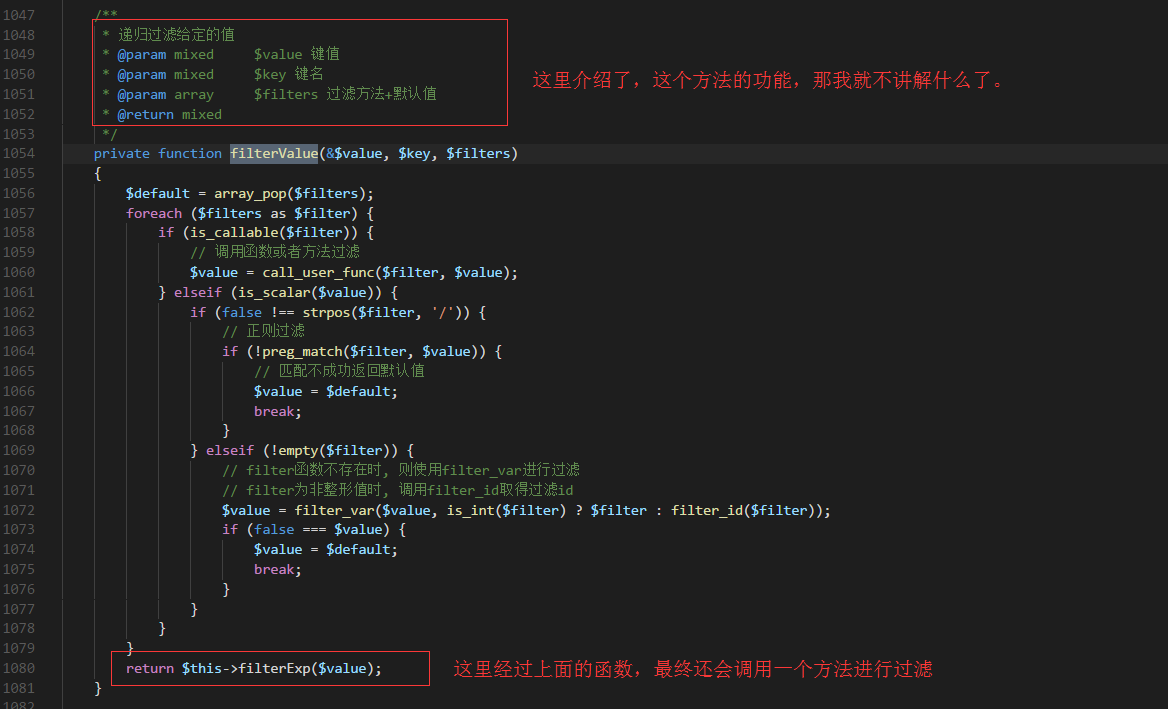
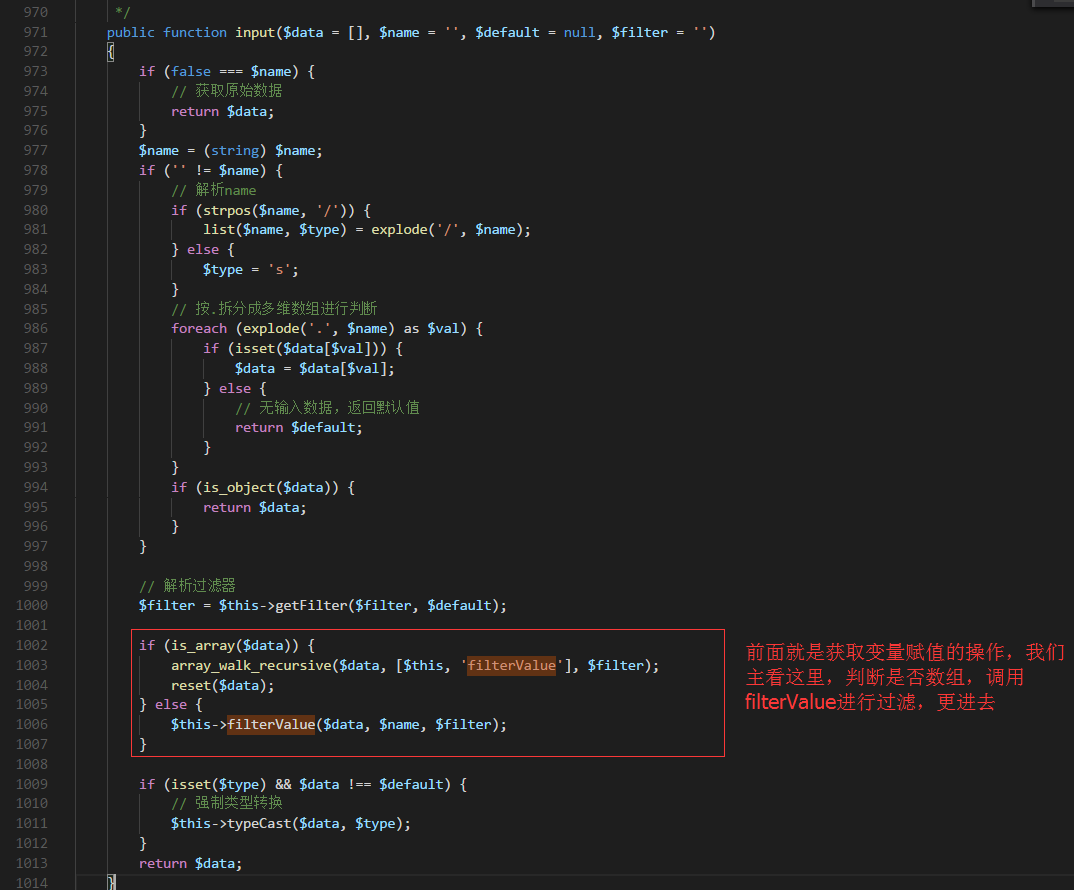




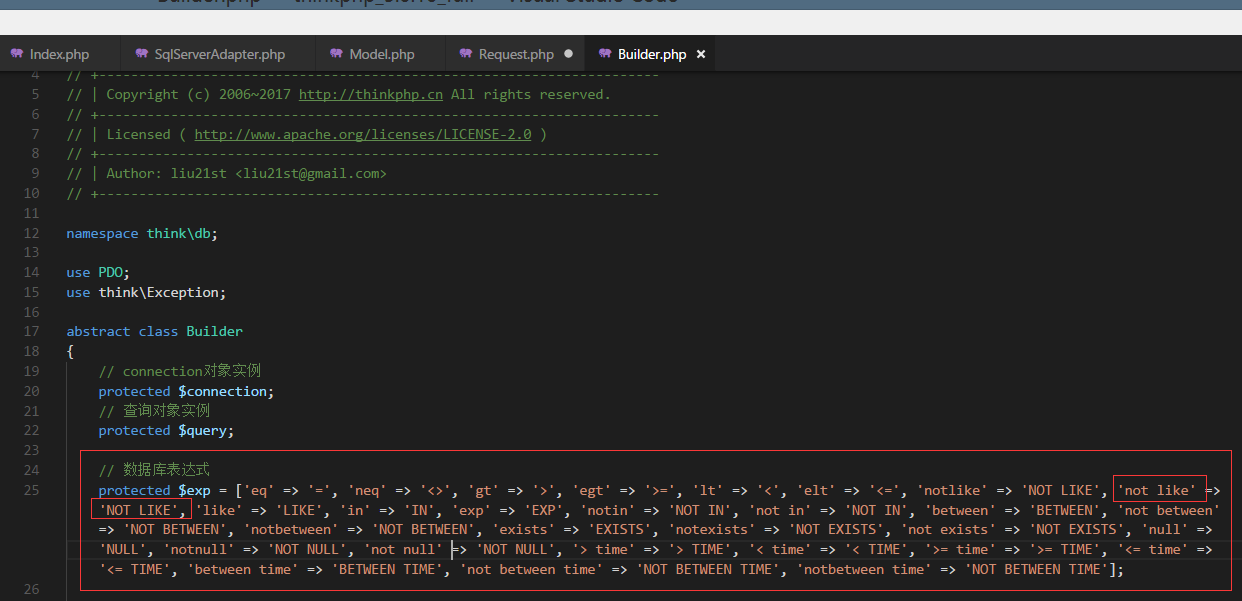
这里来看看TP系统核心安全函数之一 input函数，他可以帮助我们获取各种变量与自动过滤的功能等。

文件： thinkphp\library\think\Request.php

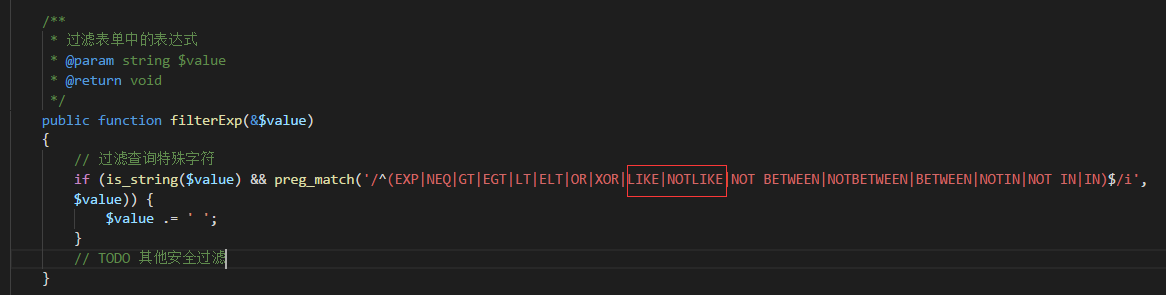
方法： input()



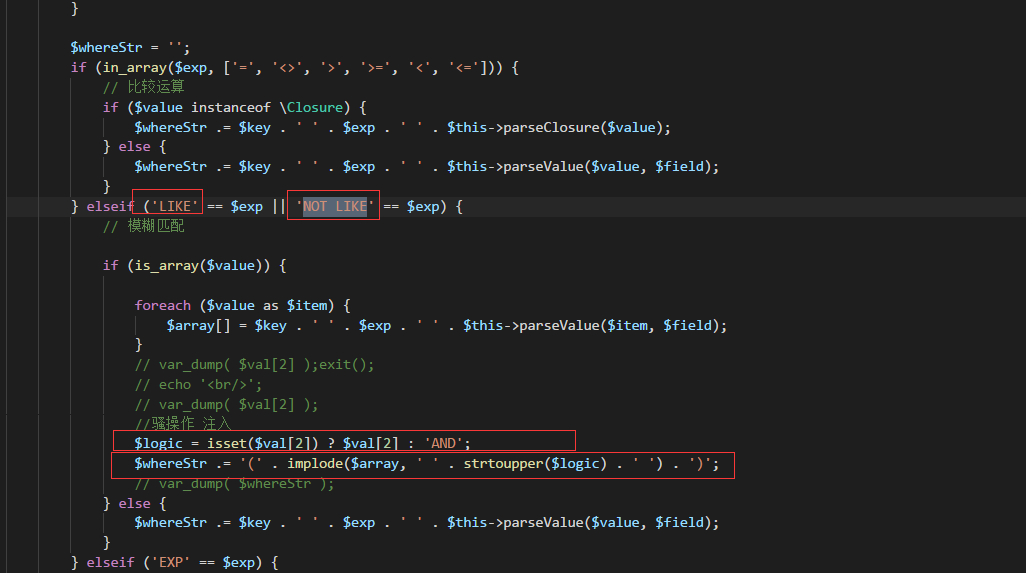
从这个函数来说，我们一般情况下就算可以控制传入的变量，也会因为多个空格导致匹配不上，但是这里因为TP5重写了数据库操作的方法，忽略了一些东西，所以导致我们可以投机取巧的绕过他。如下图



这个是TP5 支持的表达式，在看看input会进行过滤的表达式



通过对比是否就有一个比较清楚的概念了？没错TP5 新添加了一个表达式 not like 但是在 filterExp中又没有进行过滤，最终导致了可注入。



修复方法：

打开文件：\thinkphp\library\think\Request.php

打开方法：filterExp();

添加正则：/^(EXP|NEQ|GT|EGT|LT|ELT|OR|XOR|LIKE|NOTLIKE|NOT LIKE|NOT BETWEEN|NOTBETWEEN|BETWEEN|NOTIN|NOT IN|IN)$/i

