

New issue

Jump to bottom

## /sys/duplicate/check存在sql注入漏洞 #4129

🔒 Closed azraelxuemo opened this issue on Oct 25 · 10 comments

azraelxuemo commented on Oct 25

[jeecg-boot漏洞.pdf](#)

zhangdaiscott commented on Oct 30

Member

你这个是哪个版本，针对注释这种我们处理过

azraelxuemo commented on Oct 30

Author

就是最新的，但是可以绕过  
...

zhangdaiscott commented on Oct 30

Member

截图版本号

azraelxuemo commented on Oct 30

Author

你们自己看我发的内容，里面你们加checked了啊，但是check有问题，可以被bypass我已经说的很详细了，我是直接clone你们的项目  
...

zhangdaiscott commented on Oct 30

Member

改成这样就好了

```
SqlInjectionUtil.java (F:\gitcode\jeecg-boot-github\jeecg-boot-base-core\src\main\java\org\jeecg\common\util) [默认]
fb8afcd7d4fdd9cf5a2b023b46c1953e609c96b
* SQL注入以双引号处理，遇到单引号则用单引号替换

* @param values
* @return
*/
public static void filterContent(String[] values, String customXssString) {
    String[] xssArr = XSS_STR.split("\\\\");
    for (String value : values) {
        if (value == null || "".equals(value)) {
            return;
        }
        // 统一转为小写
        value = value.toLowerCase();
        //SQL注入检测存在绕过风险 https://github.com/jeecg/jeecg-boot/issues/14262
        value = value.replaceAll("'", "\\'");
        for (int i = 0; i < xssArr.length; i++) {
            if (value.indexOf(xssArr[i]) > -1) {
                Log.error("请注意，存在SQL注入关键词--> {}", xssArr[i]);
                Log.error("请注意，值可能存在SQL注入风险!--> {}", value);
                throw new RuntimeException("请注意，值可能存在SQL注入风险!-->" + value);
            }
        }
    }
}

您的版本
* SQL注入以双引号处理，遇到单引号则用单引号替换

* @param values
* @return
*/
public static void filterContent(String[] values, String customXssString) {
    String[] xssArr = XSS_STR.split("\\\\");
    for (String value : values) {
        if (value == null || "".equals(value)) {
            return;
        }
        // 统一转为小写
        value = value.toLowerCase();
        //SQL注入检测存在绕过风险 https://github.com/jeecg/jeecg-boot/issues/14262
        value = value.replaceAll("'", "\\'");
        for (int i = 0; i < xssArr.length; i++) {
            if (value.indexOf(xssArr[i]) > -1) {
                Log.error("请注意，存在SQL注入关键词--> {}", xssArr[i]);
                Log.error("请注意，值可能存在SQL注入风险!--> {}", value);
                throw new RuntimeException("请注意，值可能存在SQL注入风险!-->" + value);
            }
        }
    }
}
```

azraelxuemo commented on Oct 30

Author

okok好好的，因为我看到的是checksql可以被绕过，所以就提出来了哈哈哈2333  
...

azraelxuemo commented on Oct 30

Author

```
TemplateController.java SystemApiController.java SysAnnouncementController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3 <modelVersion>4.0.0</modelVersion>
4 <groupId>org.jeecgframework.boot</groupId>
5 <artifactId>jeecg-boot-parent</artifactId>
6 <version>3.4.3</version>
7 <packaging>pom</packaging>
8 <name>JEECG BOOT ${project.version}</name>
```

这是我的版本号

azraelxuemo commented on Oct 30 • edited

Author

我个人建议还是把sql注入里面的空格删掉  
因为你们替换了//  
但还可以用0绕过  
updatexml(1,(select(if(length("aaa")>5,1,sleep(10)))union select(1)),1)  
所以索性你们就不替换//这些  
然后直接把输入的整个字符串转成小写  
判断有没有select.这种关键字

Authorize(default) X

重置校验接口 X

文档

请试

Open

GET

/jeecg-boot/sys/duplicate/check

请求头部

请求参数

AfterScript

x-www-form-urlencoded

form-data

raw

参数名称

参数值

dataId

2000

fieldName

updatexml(1,(select(if(length("aaa")>5,1,sleep(10)))union select(1)),1)

fieldVal

1000

tableName

sys\_log

响应内容

Raw

Headers

Curl

1-7

{  
 "success": true,  
 "message": "该值可用!",  
 "code": 200,  
 "result": "该值可用!",  
 "timestamp": 1667184013649  
}

成功标志  
返回处理消息  
返回代码  
返回数据对象  
时间戳

显示说明

响应码: 200

您看这样还是可以注入的  
就算我修改了还是可以绕过的

```
value = value.toLowerCase();  
//SQL注入检测存在绕过风险 https://gitee.com/jeecg/jeecg-boot/issues/T4NZG6  
value = value.replaceAll(regex: "/\\s.*\\s/", replacement: " ");  
return findDuplicatesForTimestamp(value);
```

zhangdaiscott added a commit that referenced this issue on Nov 1

sql注入检查更加严格，修复/sys/duplicate/check存在sql注入漏洞 #4129

f18ced5

zhangdaiscott commented on Nov 1

Member

已修复

zhangdaiscott closed this as completed on Nov 1

azraelxuemo commented on Nov 1

Author

好的，辛苦啦

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

