vran-dev / databasir Public

Code Issues 9 Pull requests Discussions Actions Projects 1 ···

Databasir 1.01 has Server-Side Request Forgery vulnerability.

Moderate **vran-dev** published **GHSA-r8m9-r74j-vc6m** on Apr 18

Package

No package listed

Affected versions	Patched versions
<= 1.0.1	> 1.0.1 or latest

Description

Impact

Databasir is a team-oriented relational database model document management platform. Databasir 1.01 has Server-Side Request Forgery vulnerability.

Patches

https://github.com/vran-dev/databasir/blob/master/api/src/main/java/com/databasir/api/advice/DatabasirExceptionAdvice.java

```
 @ExceptionHandler({Exception.class})
    public ResponseEntity<Object> handleUnspecificException(Exception ex, WebRequest request) {

        String path = getPath(request);
        String errorMsg = ex.getMessage();
        log.error("Unspecific exception, request: " + path + ", exception: " + errorMsg + ":", e
        return handleNon200Response(errorMsg, HttpStatus.INTERNAL_SERVER_ERROR, path);
    }
```

Workarounds

Can will be affected by the first source (https://github.com/vran-dev/databasir/blob/master/api/src/main/java/com/databasir/api/advice/DatabasirExceptionAdvice.java), one of the 111 to 118 line commented out.

References

None

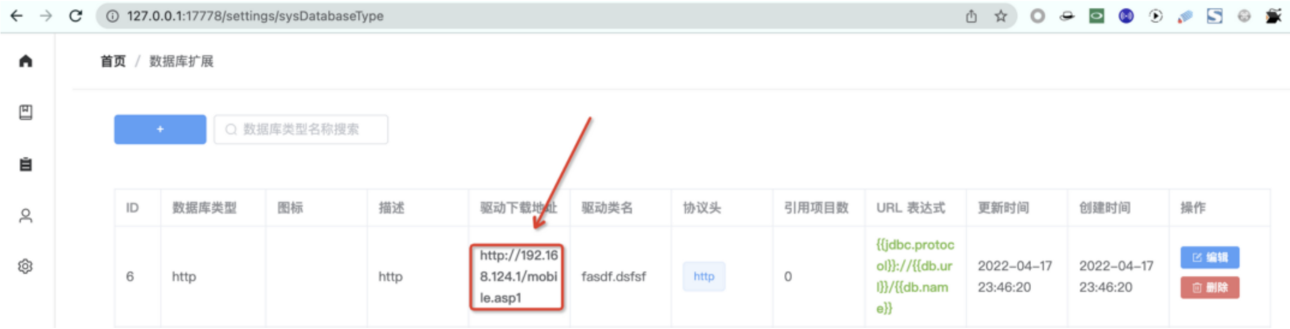
For more information

Affected source code: https://github.com/vran-dev/databasir/blob/master/api/src/main/java/com/databasir/api/advice/DatabasirExceptionAdvice.java, the vulnerability is located at the function point [database Extension -JDBC driver download address]. During the download verification process, the corresponding JDBC driver download address will be downloaded first, but this address will return a response page with complete error information when accessing a non-existent URL. Attackers can take advantage of this feature for SSRF. The vulnerability code is in lines 111 through 118:

```
@ExceptionHandler({Exception.class})
    public ResponseEntity<Object> handleUnspecificException(Exception ex, WebRequest request)
{

        String path = getPath(request);
        String errorMsg = ex.getMessage();
        log.error("Unspecific exception, request: " + path + ", exception: " + errorMsg +
":", ex);
        return handleNon200Response(errorMsg, HttpStatus.INTERNAL_SERVER_ERROR, path);
    }
```

Create a database extension named http in the [Database extension-JDBC Driver Download Address] function and fill in other Web services on the Intranet. For example: `http://192.168.124.1/mobile.asp1` .

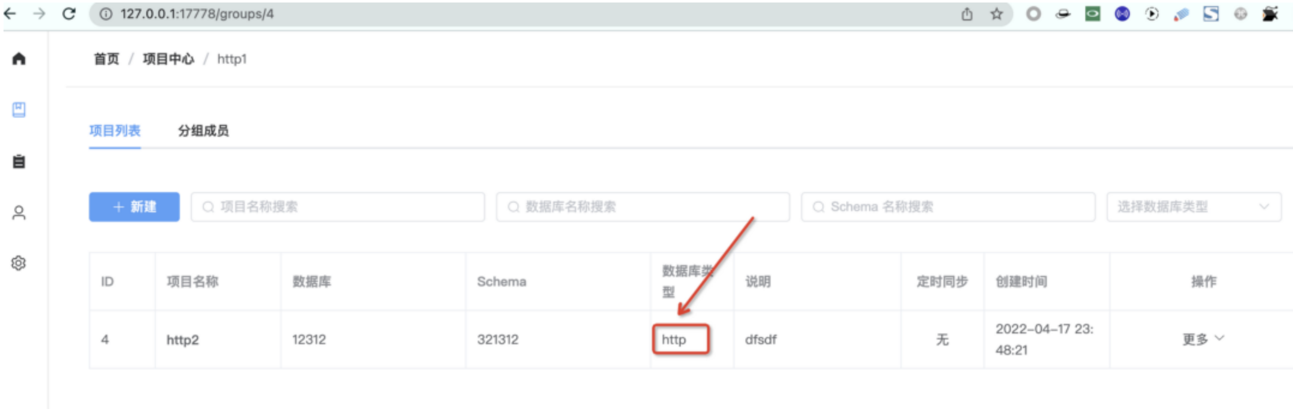


127.0.0.1:17778/settings/sysDatabaseType

首页 / 数据库扩展

ID	数据库类型	图标	描述	驱动下载地址	驱动类名	协议头	引用项目数	URL 表达式	更新时间	创建时间	操作
6	http		http	http://192.168.124.1/mobile.asp1	fasdf.dsfsf	http	0	{{jdbc.protocol}}://{{db.url}}/{{db.name}}	2022-04-17 23:46:20	2022-04-17 23:46:20	编辑 删除

Then create a new project in the project center and introduce our database extension called http.

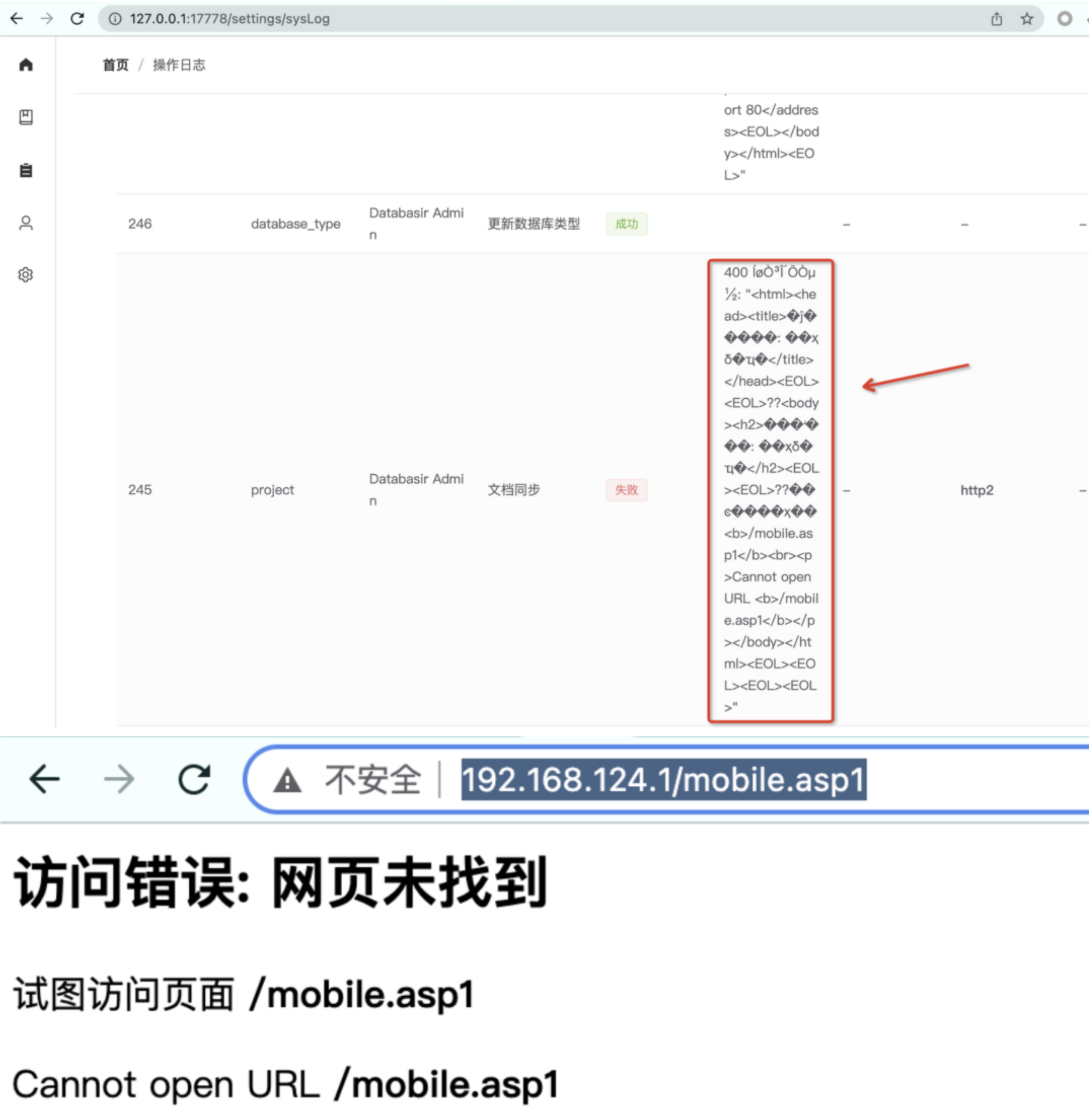


127.0.0.1:17778/groups/4

首页 / 项目中心 / http1

ID	项目名称	数据库	Schema	数据库类型	说明	定时同步	创建时间	操作
4	http2	12312	321312	http	dfsdf	无	2022-04-17 23:48:21	更多

Check the logs to see if SSRF was successfully utilized.


127.0.0.1:17778/settings/sysLog
首页 / 操作日志

					ort 80</address><EOL></body></html><EOL>"			
246	database_type	Databasir Admin	更新数据库类型	成功		–	–	–
245	project	Databasir Admin	文档同步	失败	400 ÍøÒ³Î´ÕÒµ½: "<html><head><title>�ĵ�����: ��χδ�ц�</title></head><EOL><EOL>??<body><h2>������: ��χδ�ц�</h2><EOL><EOL>??��©�����χ��<b>/mobile.asp1</b> <p>Cannot open URL <b>/mobile.asp1</b></p></body></html><EOL><EOL><EOL><EOL>"	–	http2	–

不安全 | 192.168.124.1/mobile.asp1

访问错误：网页未找到

试图访问页面 **/mobile.asp1**

Cannot open URL **/mobile.asp1**

Severity

Moderate

CVE ID

CVE-2022-24862

Weaknesses

No CWEs

Credits

LuckyT0mat0