# path manipulation

文件名后台指定。不能通过前台指定文件路径

String rName = request.getParameter("reportName");

File rFile = new File("/usr/local/apfr/reports/" + rName);

@GetMapping(value = "/demo/exportPDF")

public ResponseEntity<byte[]> exportPdf(@RequestParam String fileUrl) {

# Poor Error Handling: Return Inside Finally

将返回指令移到 finally 块之外。如果必须要 finally 块返回一个值，可以简单地将该返回值赋给一个本 。

地变量，然后在 finally 块执行完毕后返回该变量。

# Portability Flaw: File Separator

文件分隔符用File.separator

File file = new File(directoryName + "\\" + fileName);

改成

File file = new File(directoryName + File.separator + fileName);

# Portability Flaw: Locale Dependent Comparison

手动方式将区域设置

String.toUpperCase(**Locale.ENGLISH**)

# Privacy Violation

对机密信息（如客户密码或社会保障号码）不要输出出来。

例如：

String decrypt = CMCSM4Utils.decrypt(body, appKeyDTO.getSecretKey());

System.out.println(decrypt);

System.out.println(username1);

System.out.println(password1);

System.out.println(username2);

System.out.println(password2);

# Race Condition: Singleton Member Field

Servlet 为单例模式。不要把用户数据存储在 Servlet 成员字段中，这种方式会导致某个用户可能在无意中看到其他用户的数据。不要为任何参数（常量除外）使用 Servlet 成员字段，可以考虑声明一个单独的类。

例如：

public class GuestBook extends HttpServlet {

String name;

protected void doPost (HttpServletRequest req, HttpServletResponse res) {

name = req.getParameter("name");

...

out.println(name + ", thanks for visiting!");

}

}

应改为：

public class GuestBook extends HttpServlet {

protected void doPost (HttpServletRequest req, HttpServletResponse res) {

GBRequestHandler handler = new GBRequestHandler();

handler.handle(req, res);

}

}

public class GBRequestHandler {

String name;

public void handle(HttpServletRequest req, HttpServletResponse res) {

name = req.getParameter("name");

...

out.println(name + ", thanks for visiting!");

}

}

# SQL Injection

Sql操作，，查询条件组织，使用tkmybatis，需要放到dao

# Unreleased Resource: Database

首先数据库操作得放到dao层，

Jdbc操作得正常close

尽量不使用connection 等jdbc，，，最少使用jdbctemplate

# Unreleased Resource: Streams

释放 Stream的系统资源

inputStream = new FileInputStream(excelFile);

没有close，，

# Unreleased Resource: Synchronization

ReentrantLock

没有释放锁

# Weak Encryption

不要使用第三方加密，，请使用国密算法，SMUtil

# Weak Encryption: Insecure Initialization Vector

随机算法，不要使用Random，SecureRandom请使用MySecureRandom

# Weak Encryption: Insecure Mode of Operation

不要使用第三方加密，，请使用国密算法，SMUtil

ortify典型问题归纳

Access Control: Database

对需要查询的数据，给sql语句加上数据权限的限定条件，限定数据所属角色。

对于这个加上的数据权限的限定条件，最好是从后台获取，而不是通过前台传入。

不在ms层出现持久层表字段相关代码,尽量封装到dao层

Access Specifier Manipulation

Access Specifier Manipulation解决方案（Spring）\_java\_zk052300的博客-CSDN博客

Cookie Security: Overly Broad Path

不允许将Cookie.setPath设置为"/"

Dynamic Code Evaluation: Unsafe Deserialization

移除此依赖

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

</dependencies>

Header Manipulation

使用工具类取Header中的值,过滤非法字符

com.sgcc.eip.cmc.suite.starter.webmvc.util.HttpHeadersUtil

Header Manipulation: SMTP

Insecure Randomness

使用com.sgcc.eip.cmc.suite.starter.global.util.MyRandomUtil

Insecure SSL: Overly Broad Certificate Trust

Insecure SSL: Server Identity Verification Disabled

Insecure Transport: Mail Transmission

JSON Injection

将redis操作封装

Key Management: Hardcoded Encryption Key

不要在代码中写死秘钥,密码等信息

Log Forging

不直接使用slf4j的LoggerFactory,而是使用架构组封装的

com.sgcc.eip.cmc.suite.common.log.LoggerFactory

Mass Assignment: Insecure Binder Configuration

在controller中添加代码

@InitBinder()

public void initBinder(WebDataBinder binder) {

binder.setDisallowedFields(new String[]{});

}

Null Dereference

完善空判断,在间接引用可能为 null 值的对象之前，请务必仔细检查。

Often Misused: File Upload

如果可以避免上传文件，请不要允许此操作。

如果程序必须允许文件上传，则应当只接受程序需要的特定类 型的内容，从而阻止攻击者提供恶意内容

Password Management: Hardcoded Password

Path Manipulation

Poor Error Handling: Return Inside Finally

Portability Flaw: File Separator

Portability Flaw: Locale Dependent Comparison

对可能与区域设置相关的数据进行比较时，应指定相应的区域设置。如:toUpperCase(Locale.ENGLISH)

或者使用函数 java.lang.String.equalsIgnoreCase() 以防止出现此问题

Privacy Violation

机密信息（如客户密码或社会保障号码,密码等）,不要打印出来

例如:

String decrypt = CMCSM4Utils.decrypt(body, appKeyDTO.getSecretKey());

System.out.println(decrypt);

Race Condition: Singleton Member Field

SQL Injection

不能从java中直接拼接SQL,以避免SQL 注入问题

封装持久层相关代码至dao层

反例:

ResultSet fieldInfoRs = connection.createStatement().executeQuery(" SELECT TABLE\_SCHEMA databaseName,\nTABLE\_NAME tableName,\nCOLUMN\_NAME columnName,\nDATA\_TYPE dataType, \nCOLUMN\_TYPE columnType,\nCHARACTER\_MAXIMUM\_LENGTH characterMaximumLength,\nCOLUMN\_COMMENT columnComment, IS\_NULLABLE nullAble, EXTRA extra ,COLUMN\_KEY columnKey FROM information\_schema.columns WHERE table\_schema ='" + databaseName + "' AND table\_name ='" + datasourceInfoTableName + "';\n");

Unreleased Resource: Database

手动创建的数据库连接要在finally块中释放