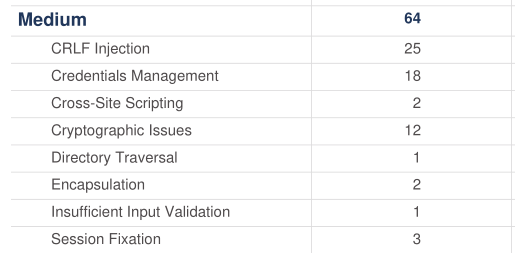
# 中级问题

## 问题概述

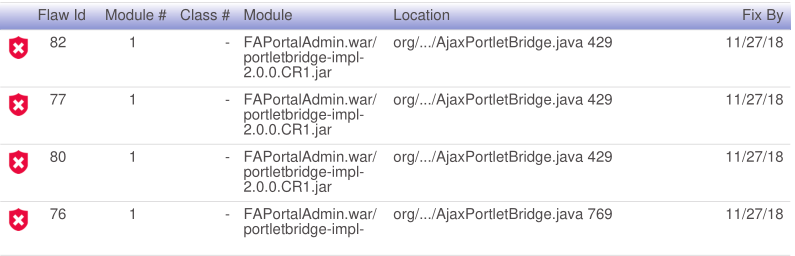


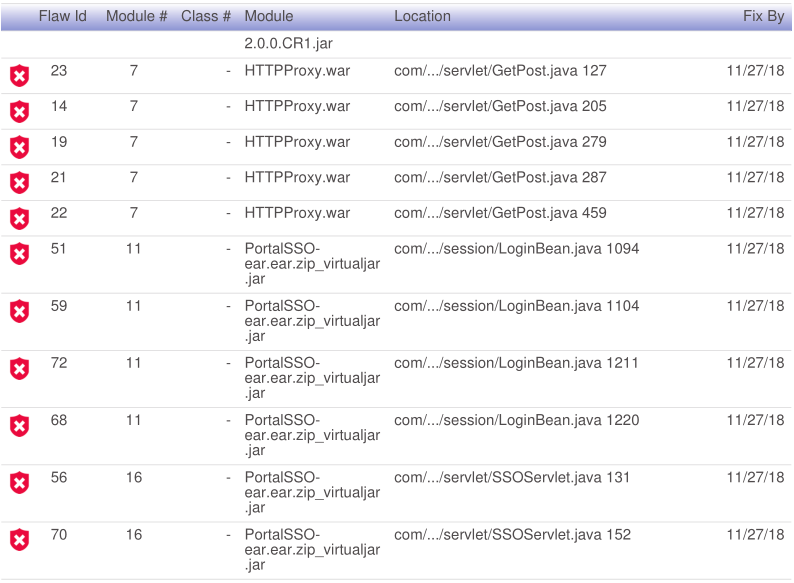
64个中级问题

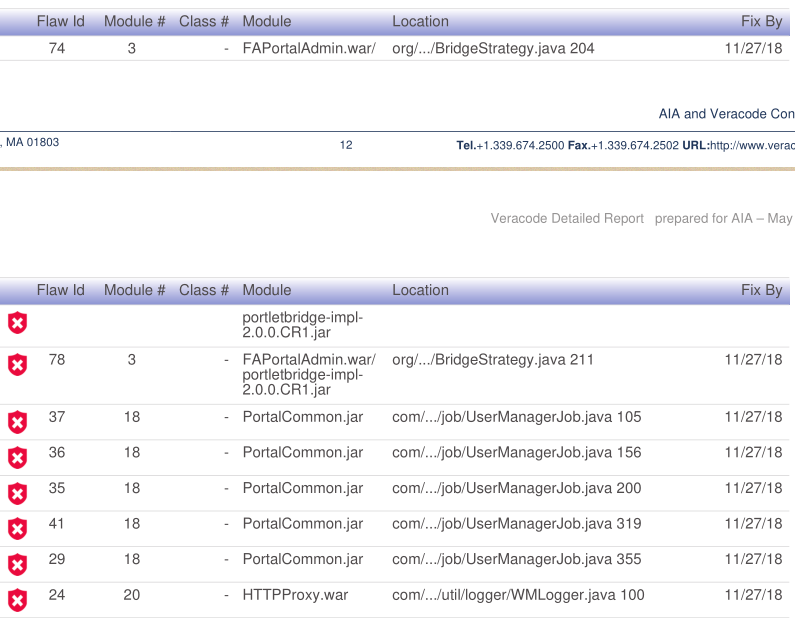
56个问题是我们自己模块中的代码问题，8个问题是Jboss框架中的问题，我们不能修改

## CRLF Injection（25）









19个我们自己的模块去修改

### 描述

首字母缩略词CRLF代表“回车，换行”，指的是用于表示文本行结尾的字符序列。当数据从不受信任的来源进入应用程序并且在使用之前未正确验证时会发生CRLF注入漏洞。例如如果攻击者能够将CRLF注入日志文件，他可能会附加伪造的日志条目，从而误导管理员或掩盖攻击的痕迹。如果攻击者能够将CRLF注入HTTP响应头，他可以使用此功能执行其他攻击，例如缓存中毒。CRLF漏洞主要影响数据完整性。

### 修复

在可能的情况下使用集中数据验证例程，对所有用户提供的数据应用强大的输入过滤。使用输出过滤器清理从用户提供的输入派生的所有输出，将非字母数字字符替换为其HTML实体等效项。

### CRLF序列的中和不当（'CRLF注射'）（2个问题）

#### 描述

函数调用包含CRLF注入缺陷。将不受信任的输入写入接口或外部应用程序，将CRLF（回车换行）序列视为用于分隔行或记录的分隔符可能导致该数据被误解。FTP和SMTP是在解析命令时将CRLF视为分隔符的协议示例。

#### 修复

当数据传递给可能错误解释它的实体时，从不受信任的输入清除CRLF序列。

#### 静态扫描发现的实例



### HTTP Header中CRLF序列的不正确中和（'HTTP响应拆分'）（15个问题）

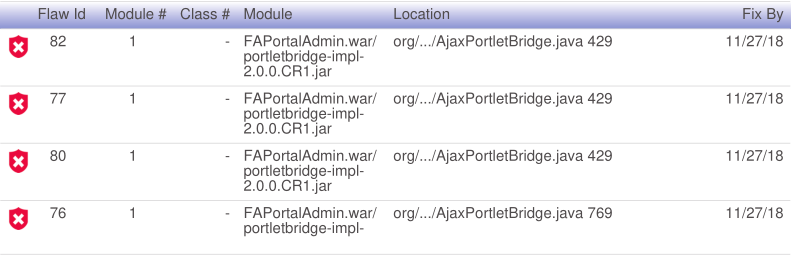
#### 描述

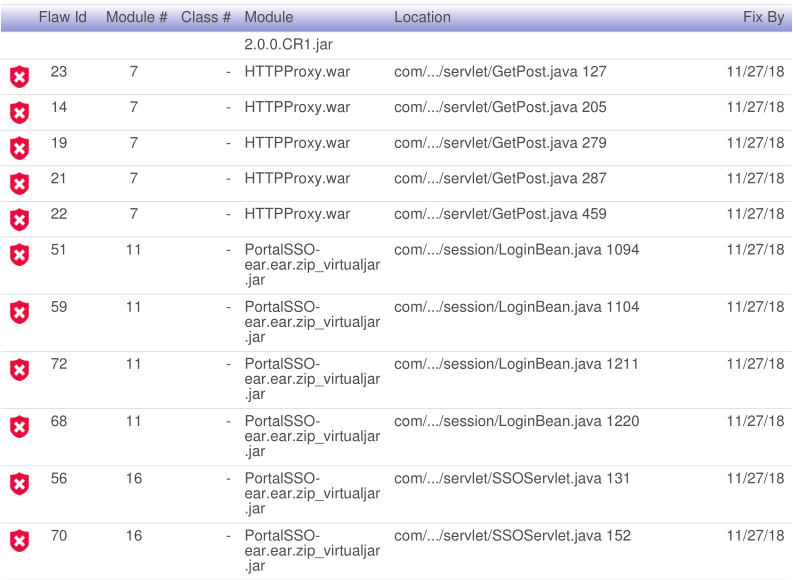
函数调用包含HTTP响应拆分缺陷。 将不受信任的输入写入HTTP标头允许攻击者操纵浏览器呈现的HTTP响应，从而导致缓存中毒和跨站点脚本攻击

#### 修复

从用于构造HTTP响应的不可信数据中删除意外的回车符和换行符。 始终验证不受信任的输入，以确保它符合预期的格式，尽可能使用集中数据验证例程

#### 静态扫描发现的实例





### 日志的输出中和不正确（8个问题）

#### 描述

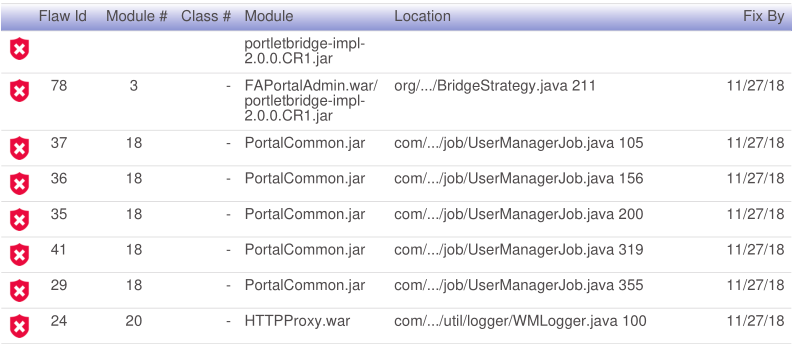
函数调用可能导致日志伪造攻击。 将不受信任的数据写入日志文件允许攻击者伪造日志条目或将恶意内容注入日志文件。 损坏的日志文件可用于覆盖攻击者的跟踪或作为攻击日志查看或处理实用程序的传递机制。 例如，如果Web管理员使用基于浏览器的实用程序来查看日志，则可能会发生跨站点脚本攻击。

#### 修复

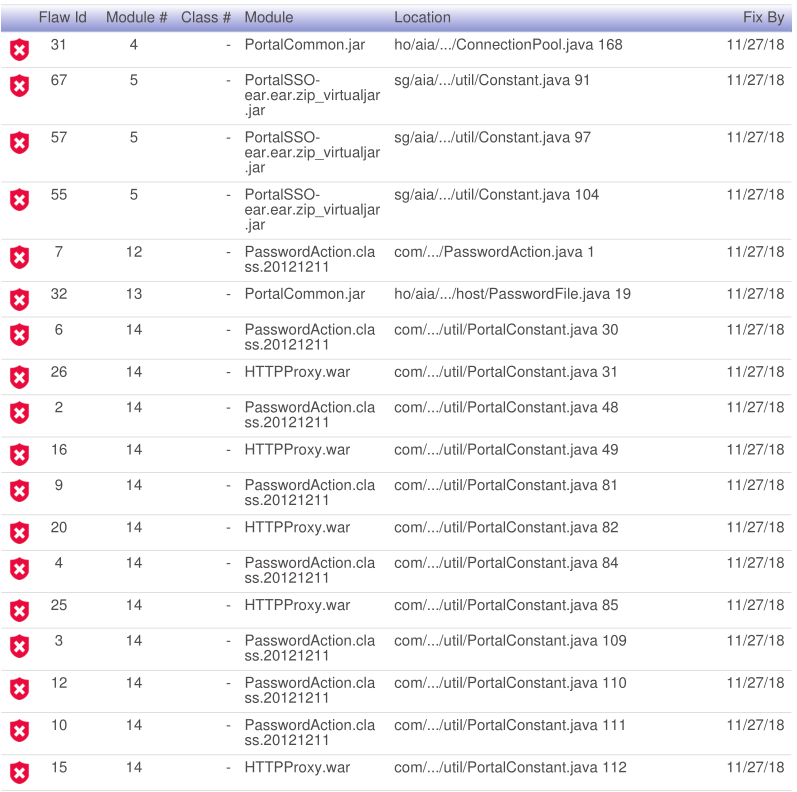
尽可能避免将用户输入直接嵌入日志文件中。 使用安全日志记录机制（例如OWASP ESAPI Logger）清理用于构造日志条目的不可信数据，该机制将自动删除意外的回车符和换行符，并且可以配置为对非字母数字数据使用HTML实体编码。 只在绝对必要时编写自定义黑名单代码。 始终验证不受信任的输入，以确保它符合预期的格式，尽可能使用集中数据验证例程。

#### 静态扫描发现的实例





## Credentials Management（18）



18个都是我们自己的模块

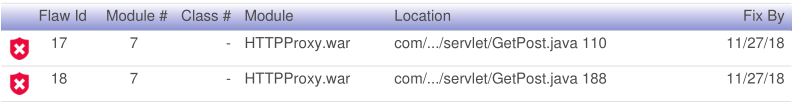
### 描述

不正确的凭据管理（例如用户名和密码）可能会危及系统安全性。 特别是，将密码以明文形式存储或将密码硬编码直接存储到应用程序代码中是一些无法轻易纠正的设计问题。 嵌入密码不仅允许所有项目的开发人员查看密码，而且还使得解决问题非常困难。 代码生产后，如果不修补软件，则无法更改密码。 如果商业产品中的硬编码密码遭到破坏，则所有部署的实例都可能容易受到攻击，从而使客户面临风险。

### 修复

避免将密码存储在易于访问的位置，并且绝不要以明文形式存储任何类型的敏感数据。 尽可能避免使用硬编码的用户名，密码或散列常量，尤其是与安全关键组件相关的问题。 从应用程序代码中带外存储密码。 遵循保护备份位置（如配置或属性文件）中存储的凭据的最佳实践

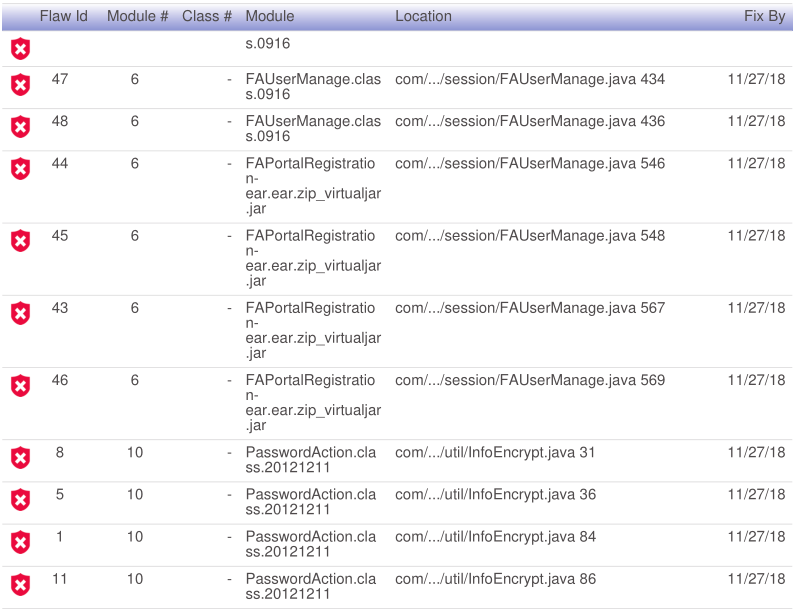
## Cross-Site Scripting（2）



2个都是我们自己的模块

## Cryptographic Issues（12）



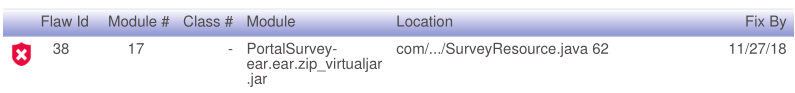


12个问题都是我们自己的模块

### 描述

### 修复

## Directory Traversal（1）



1个问题是我们自己的模块

## Encapsulation（2）





1个问题是我们自己的模块

## Insufficient Input Validation（1）



## Session Fixation（3）



3个问题是我们自己的模块