Description: CSE 5382 Secure Programming Assignment 10

Purpose: To understand and use Static Code Analysis Tools.

Part 1:

Manual Analysis:

1) Null Reference in line 58 for String command in 'SimpleWebServer.java' file provided in the assignment.

   String command = null;
2) Null Reference in line 59 for String pathname in 'SimpleWebServer.java' file provided in the assignment.

   String pathname = null;
3) Null Reference in line 87 for FileReader fr in 'SimpleWebServer.java' file provided in the assignment.

   FileReader fr=null;
4) Opening a file specified by user without any additional checks in line 103 in 'SimpleWebServer.java' file

Tool Choices/Versions:

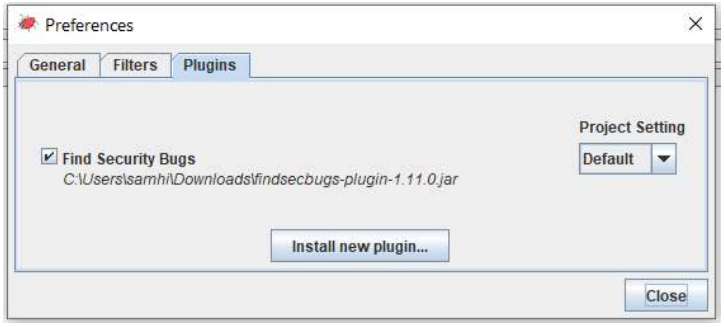
- Spotbugs 4.2.3
- Find Security Bugs Plugin Version :1.11.0
- Reshift Security IntelliJ plugin 2.5.1

Tools Invocation Process:

Spotbugs:

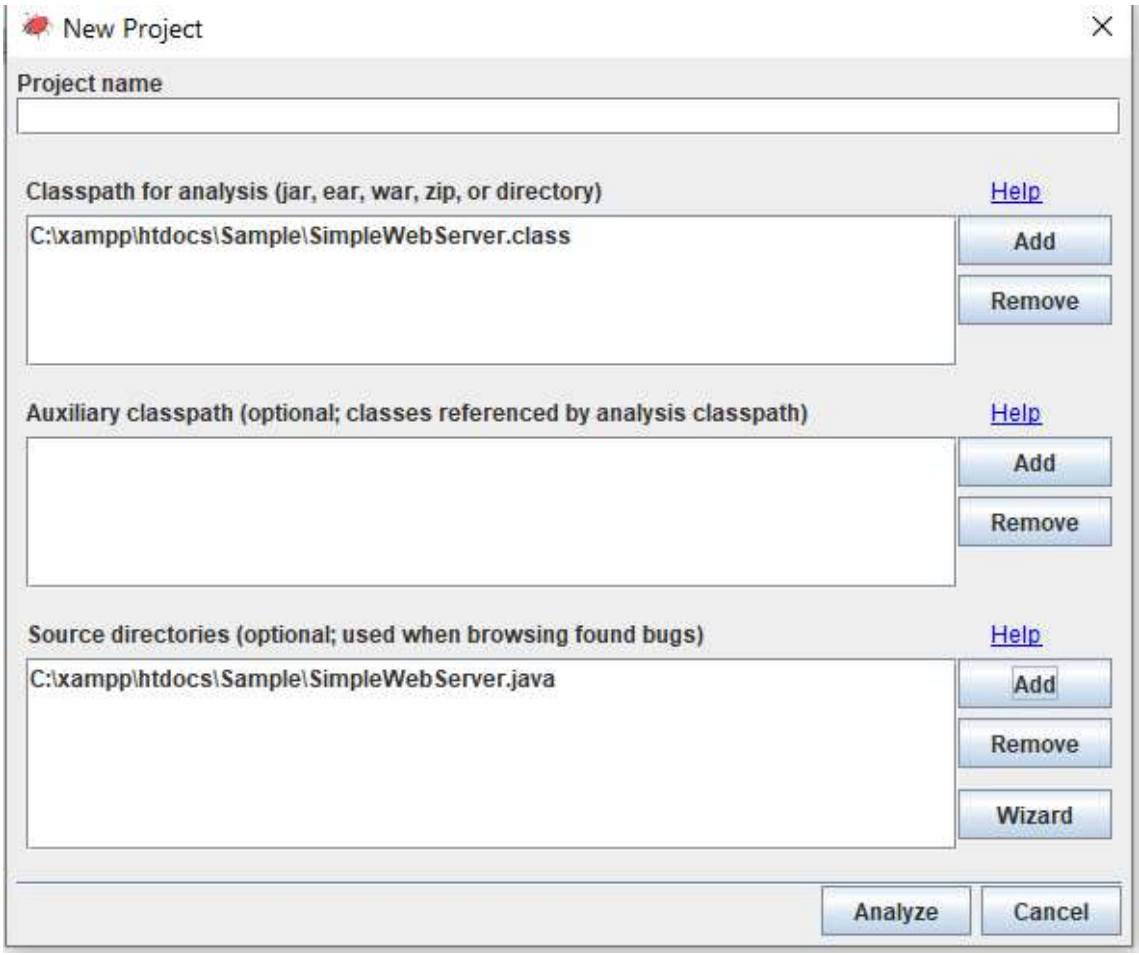
Initially the SimpleWebServer.java program is compiled. Then Spotbugs 4.2.3 tool is downloaded. Along with it find security bugs plugin version 1.11.0 is downloaded.

In this assignment spotbugs is used as stand-alone tool with find security bugs plugin enabled as shown in below screenshot.



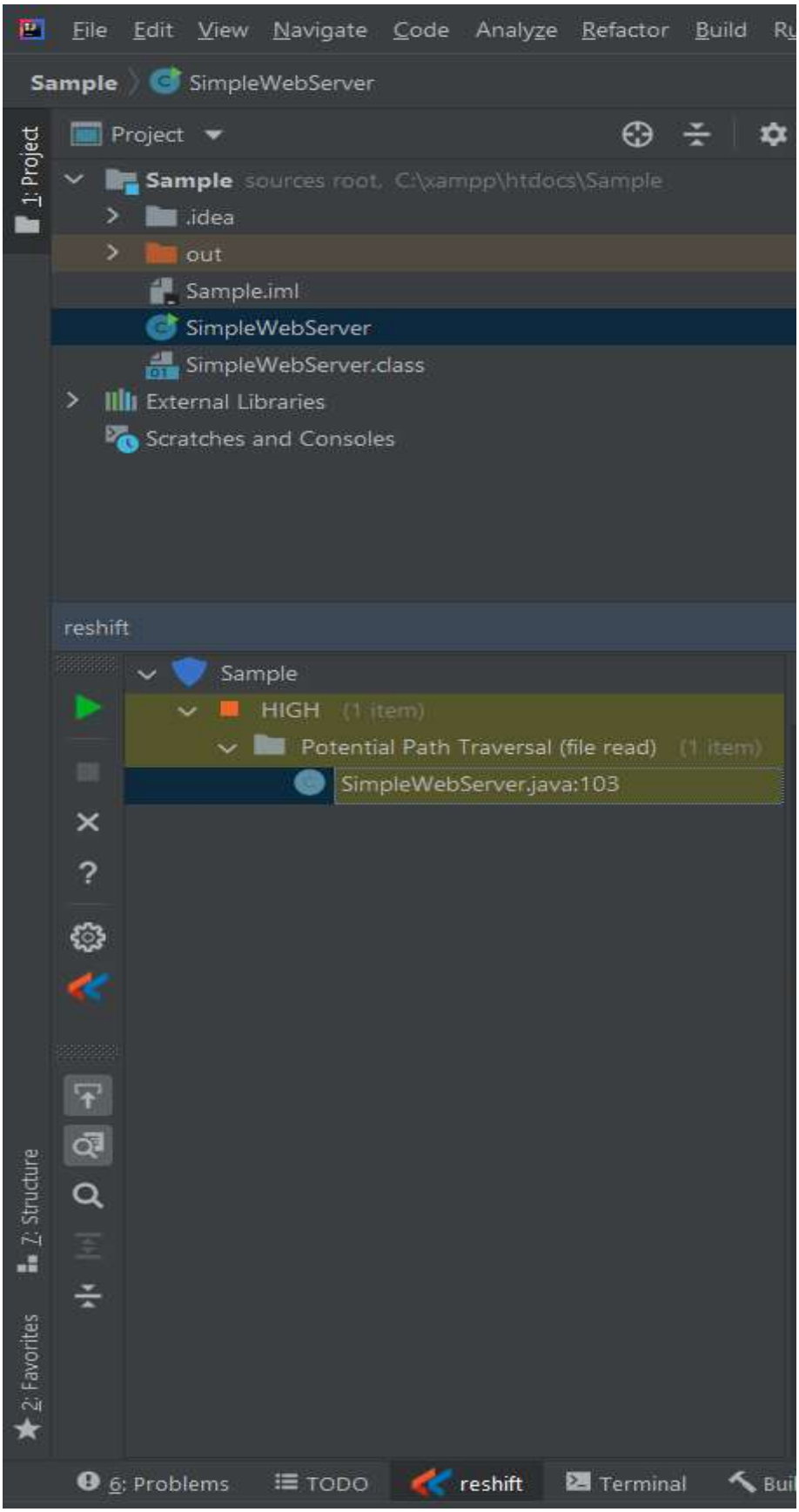
Then a new project is selected, and the Java Source File and the class file are selected as shown below for analysis as shown in below screenshot. On clicking

on analysis, we will get the analysis reports. In this way Spotbugs tool can be invoked.



Reshift Security IntelliJ plugin:

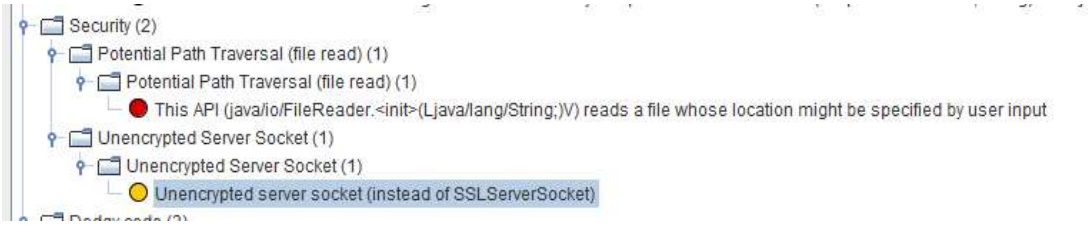
Enabled Reshift Security IntelliJ plugin 2.5.1. Then recompiled the SimpleWebServer.java program using Intellij IDE, then scanned the project using Reshift plugin to get the analysis reports.


File Edit View Navigate Code Analyze Refactor Build
Sample SimpleWebServer
1: Project
Project
Sample sources root, C:\xampp\htdocs\Sample
.idea
out
Sample.iml
SimpleWebServer
SimpleWebServer.class
External Libraries
Scratches and Consoles
reshift
Sample
HIGH (1 item)
Potential Path Traversal (file read) (1 item)
SimpleWebServer.java:103
7: Structure
2: Favorites
6: Problems
TODO
reshift
Terminal

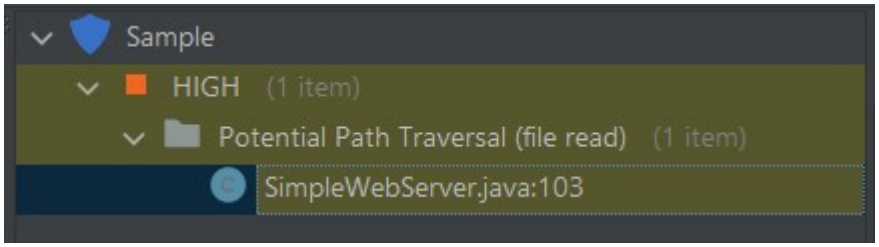
Comparison and contrast Tools:

- Both tools take source file and class file for analysis.
- Reshift tool helps in analysis of Command Injection, XPath Injection, SQL Injection, Cryptography weaknesses, etc. Software as a Service (SaaS) with ability to integrate into GitHub and other code repositories. While Spotbugs Checks for more than 400 bug patterns, including XSS, HTTP response splitting, path traversal, hardcoded password, Null dereference, etc . With find security bugs plugin to spotbugs helps in more security detectors (Command Injection, XPath Injection, SQL/HQL Injection, Cryptography weakness and many more).
- Reshift will be categorized as Security Review tool. While Spotbugs with find security bugs plugin will be categorized as Program understanding, Program verification, Property checking, Bug finding, Security review tool.
- "Unencrypted Server Socket" issue is noticed by Spotbugs tool with find Security bugs plugin enabled tool alone.
- "Potential Path Traversal" issue is noticed by both the tools.

  Spotbugs with Find security bugs plugin enabled:



  Reshift :



  - Unencrypted server socket flaw, spotbugs with find security bug reported true negative, while reshift reported false positive.

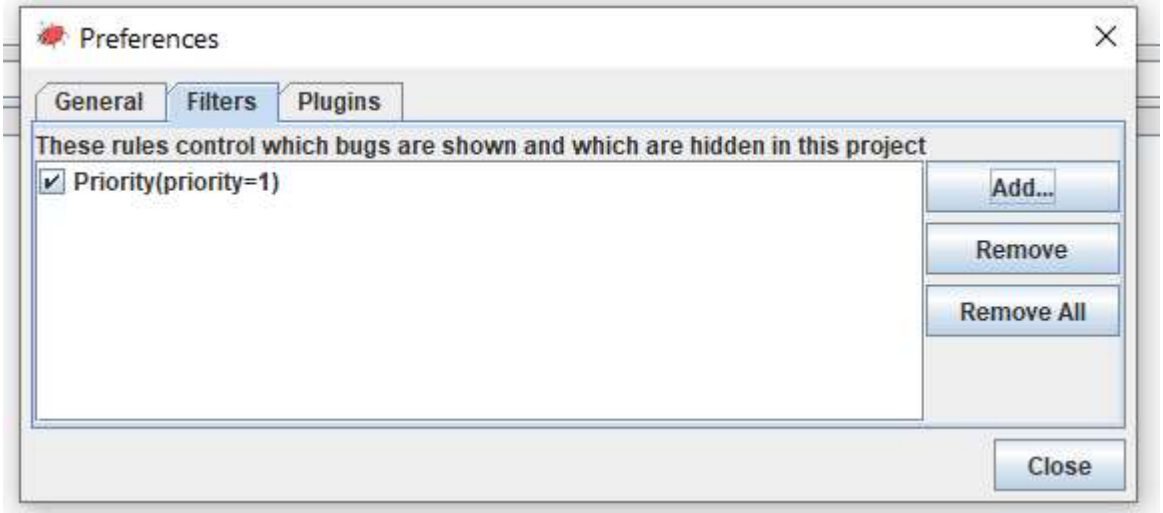
Results:

Spotbugs with find security plugin enabled:

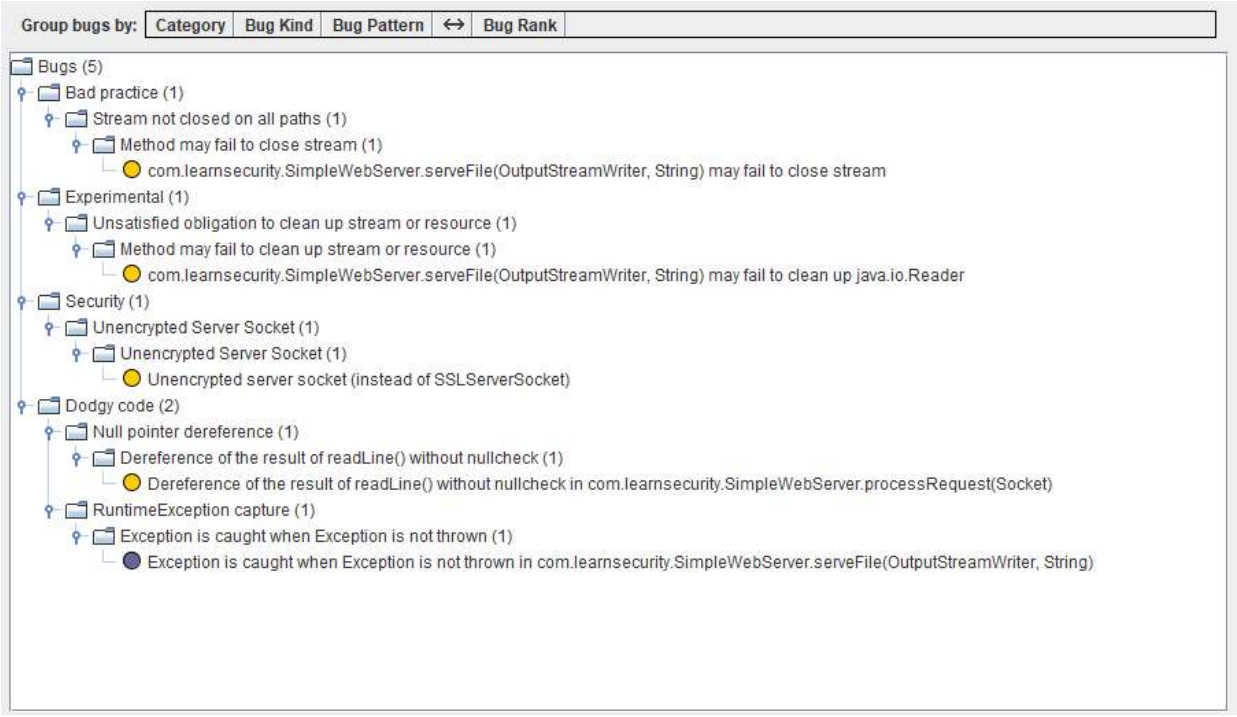
The entire analysis report is saved as "SampleWebServer_analysis.html" and will be shared along with this report.

Checked for different priority bugs by enabling the priority filter as shown in the following screenshots.

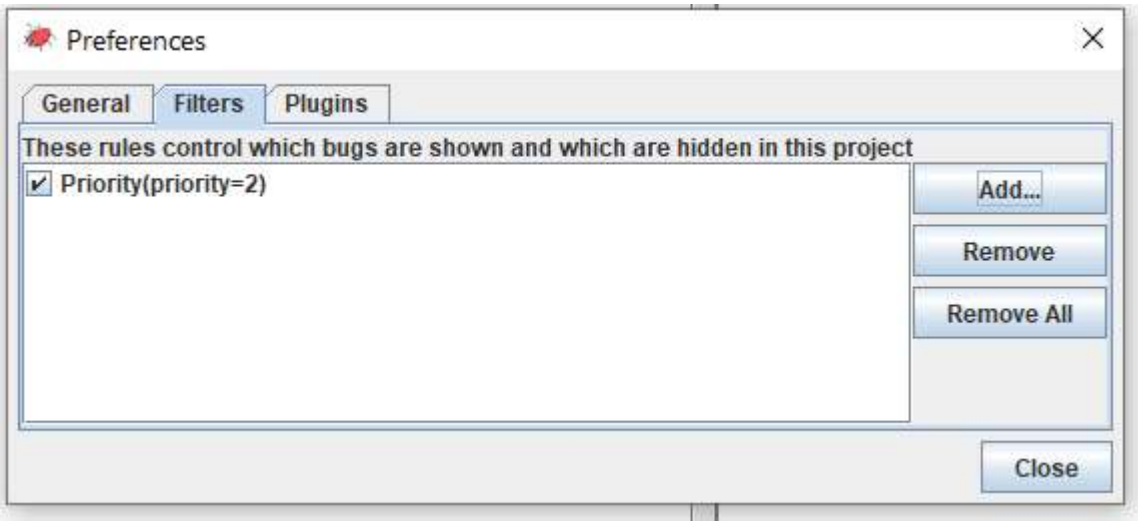
Priority filter 1 is applied:



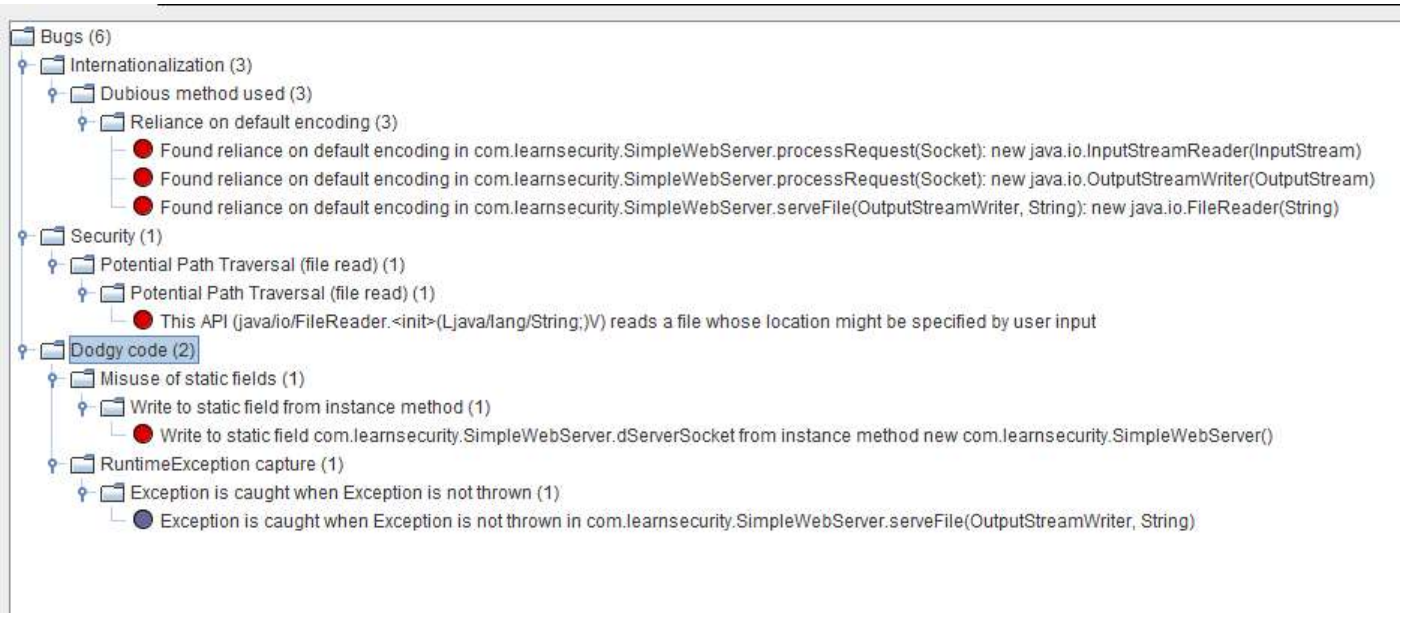
Bugs identified on applying priority filter with value 1 (high):



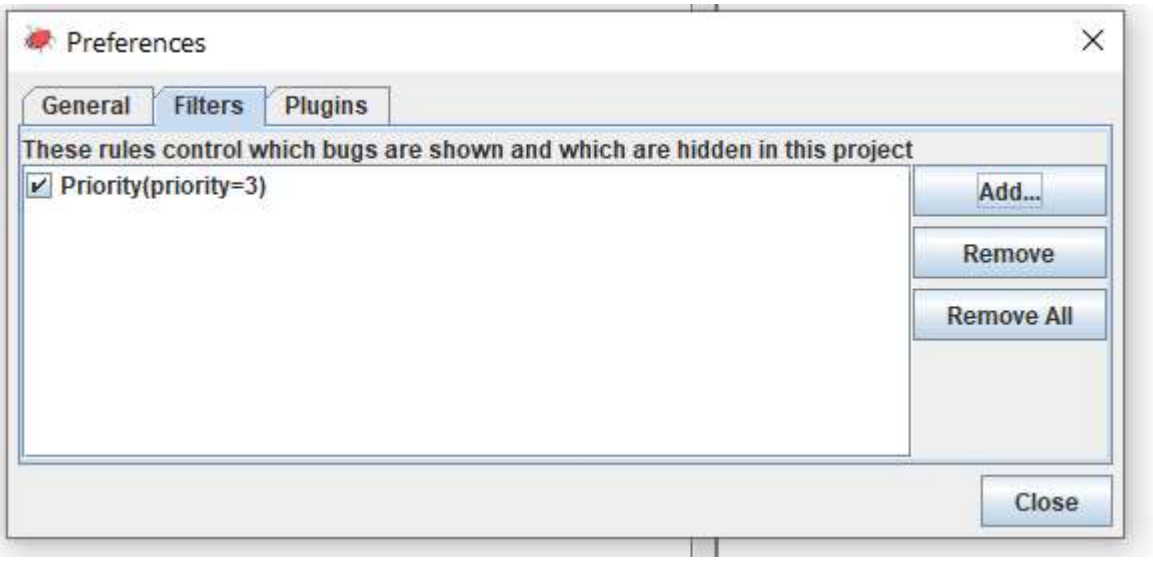
Priority Filter 2(medium) is applied:



Bugs identified on applying priority filter with value 2(medium):

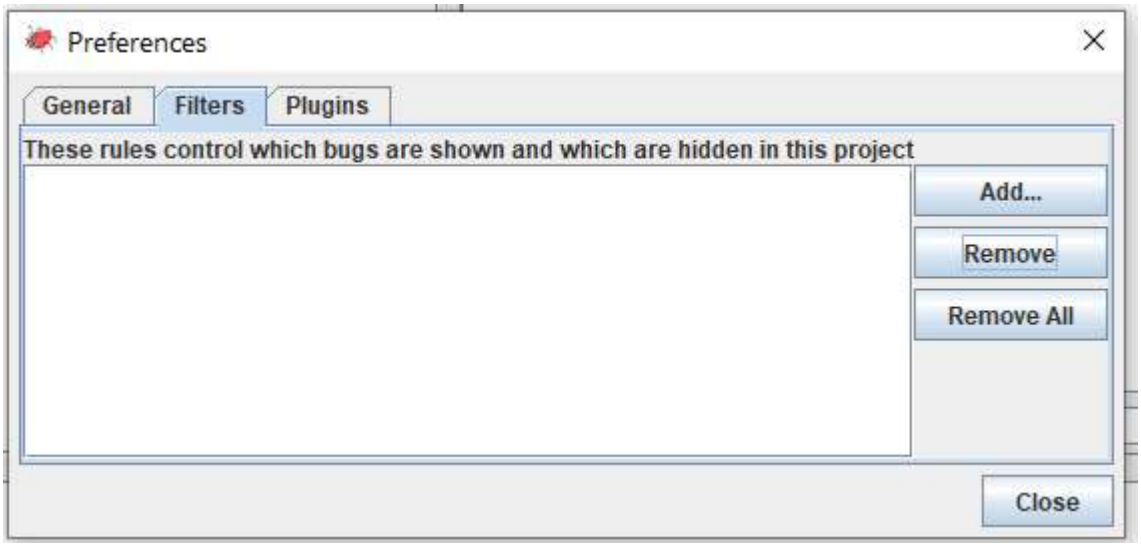


Priority filter 3(low) is applied:

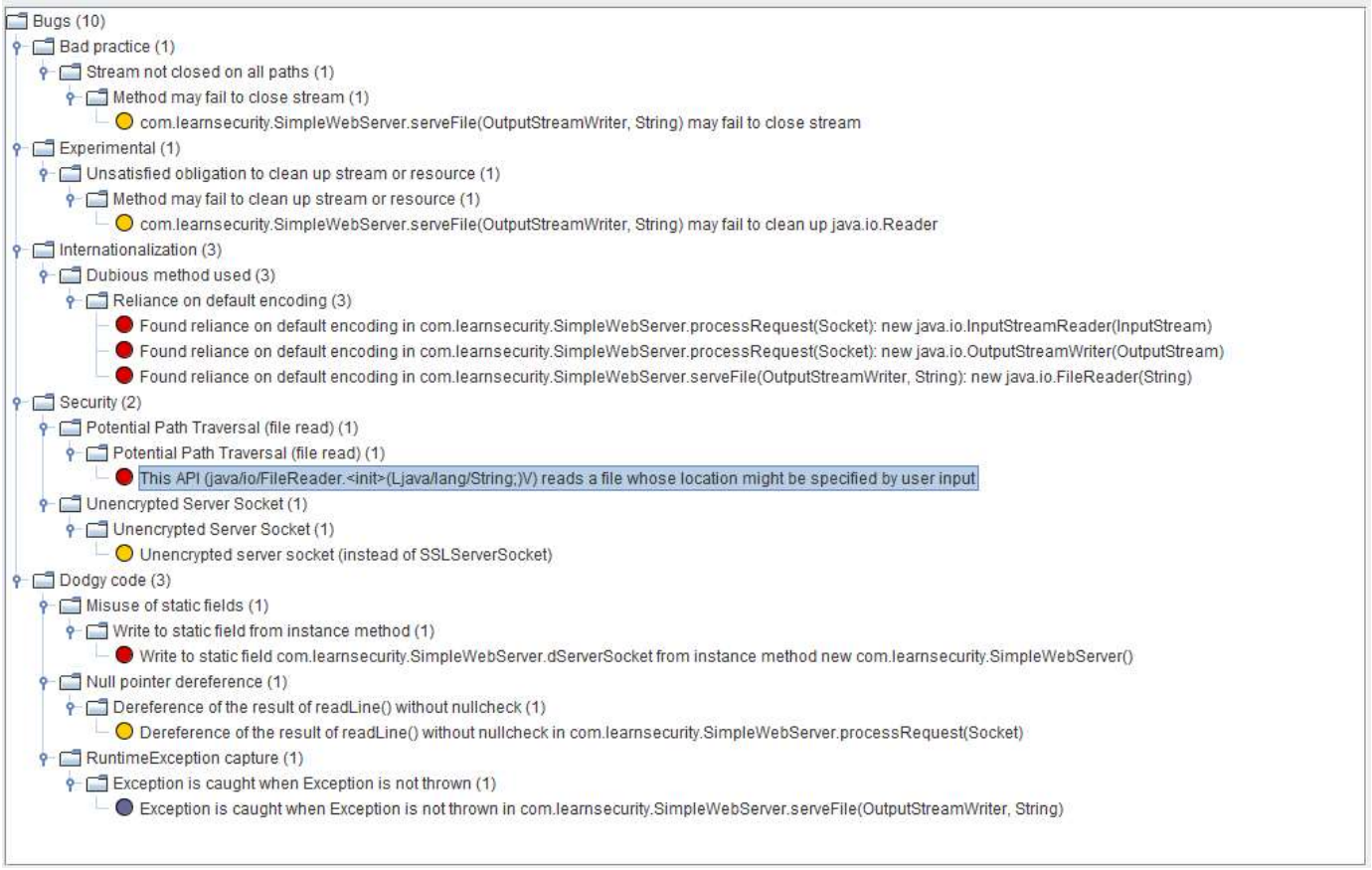


Bugs identified on applying priority filter 3(low):

No priority filter is applied:

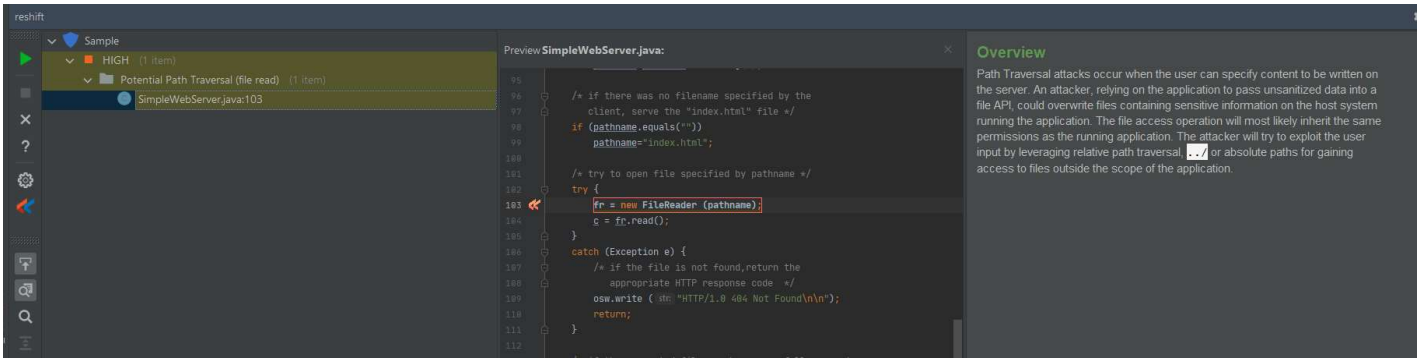


Bugs identified on applying no filters:



Reshift Security Intellij plugin Results:

The results cannot be exported, so providing the screenshot of analysis report.



Part2:

Attaching the code in the submission zip file. Analyzed Xpath.java program as a part of this.

Manual Analysis:

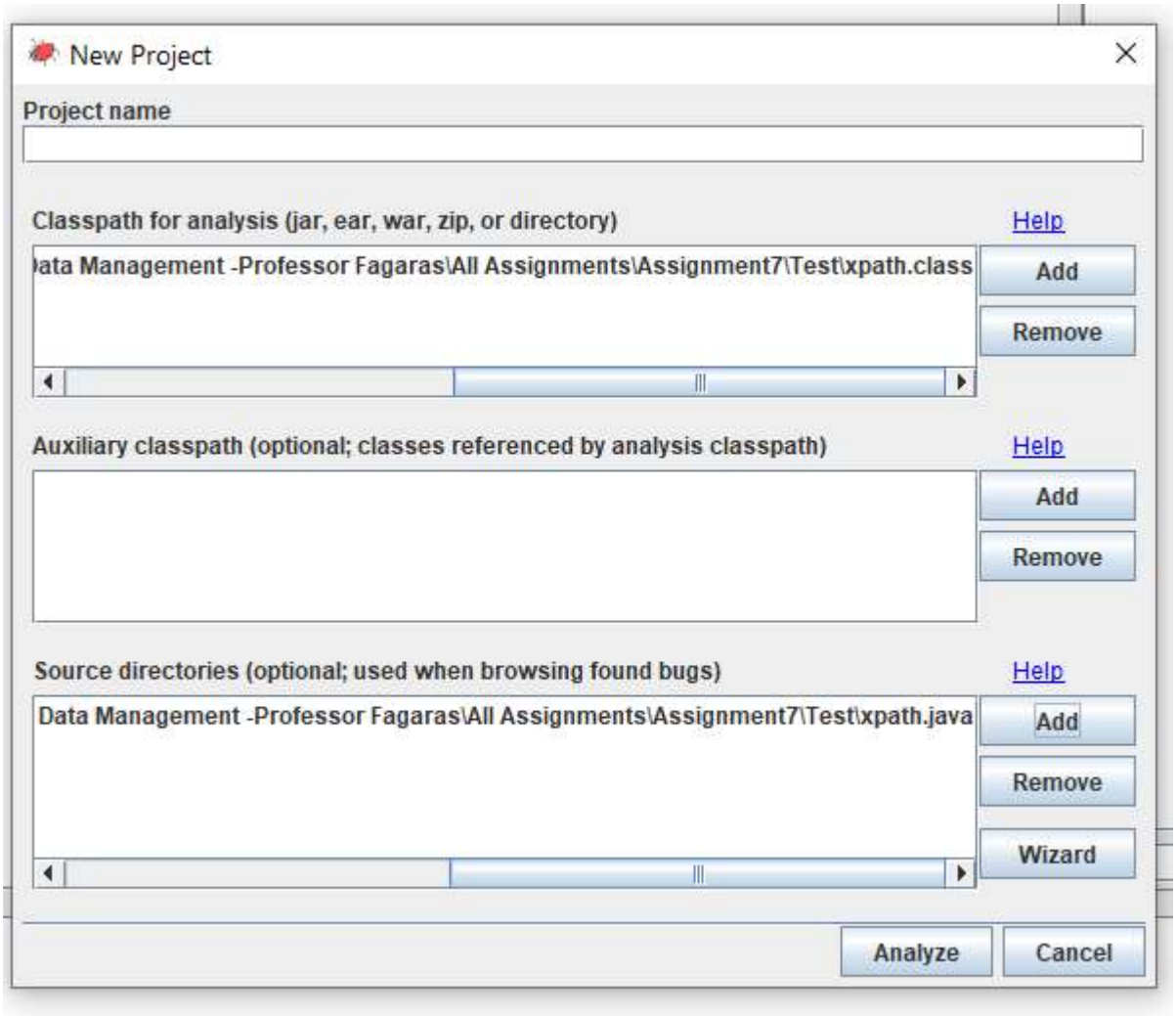
I did not identify any defects as a part of manual analysis as I am running the program on the DOM file provided by me. Potentially passing the DOM file can pose a threat but as I am just traversing the DOM tree to show xquery results, I did not find as a possible threat.

Results:

Spotbug tool with find security plugin enabled:

In addition to the screenshots, a xpath_program_analysis.html is shared along with submission of this report.

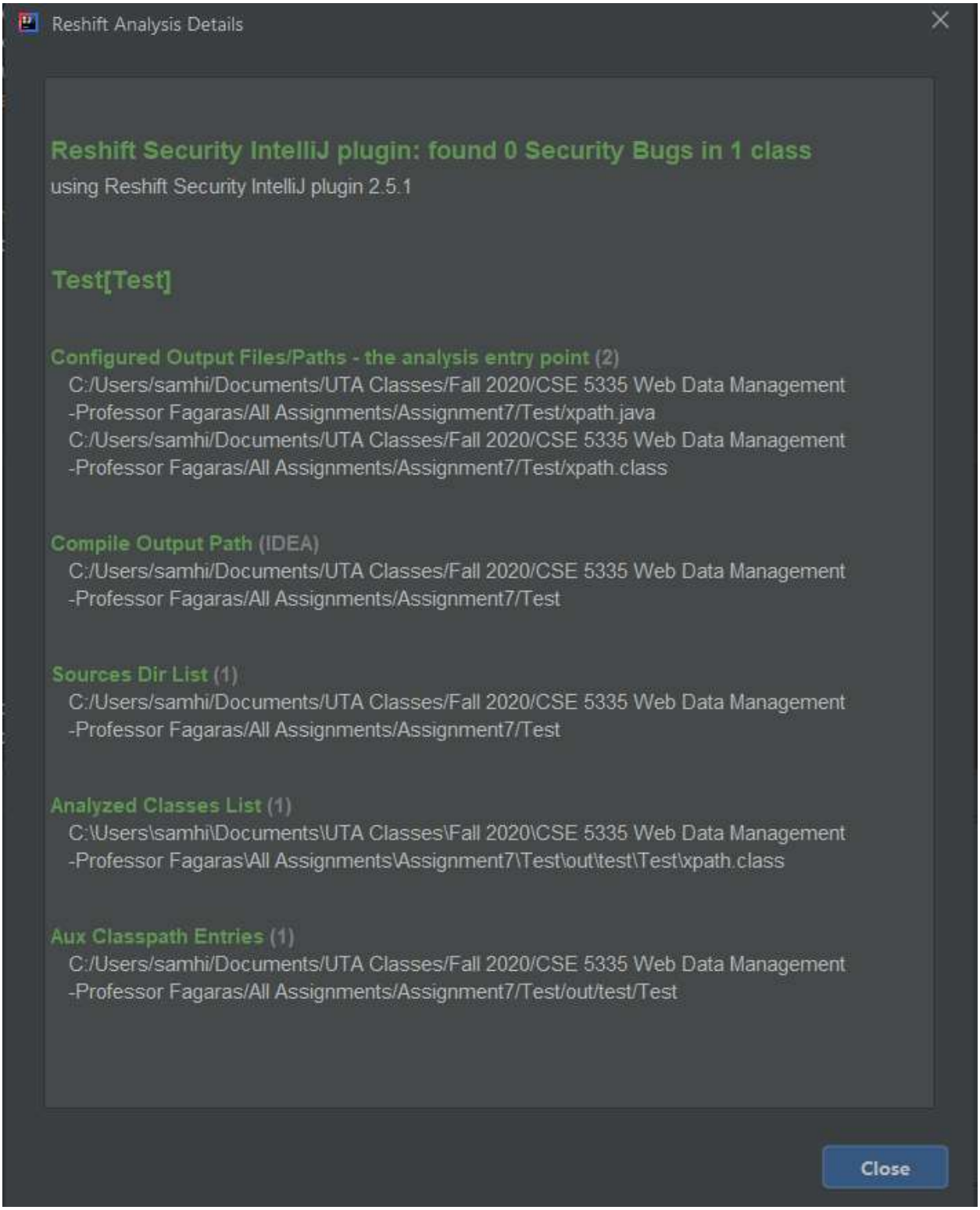
Selected the source and class files:



Analysis:

Reshift results:

As results cannot be exported, sharing the screenshots as below. No potential threats are identified in the file that is being checked.



Fixes:

No fixes are required as it is a program to show the xquery results after traversing DOM. The DOM file is shared by me. If another DOM file is shared, then also it will try to parse the DOM tree to evaluate the XQuery which will not create a potential threat as there is no sensitive information available in parsing the DOM tree.

References:

- Source Code Security Analyzers - SAMATE (nist.gov)
- Assignment sheet provided.
- Secure programming Class lecture