Part 1:

Manual Analysis:



Final can be added in front of the code to clear the warning, this warning is found by eclipse before execution of the program.



Included: FileReader that is declared and opened here is not closed in the program. So we need to add fr.close() at the end of the code.

Legitimate finding: the prgm code which is inside of the main class is not written in the try-catch block. A character array usage would be better then strings used in the program which can decrease program vulnerability.

Tool Choices/Versions :

* Windows 11
* SporBugs eclipser plugin

Graphical user interface, text, application, email

Description automatically generated

* Pmd eclipse pluguin

Graphical user interface, text, application

Description automatically generated

Tool Invocation Process:

We use the latest version of eclipse

The tools/ plugins are installed from the eclipse market place. We search for spotbugs and click on install and the ide prompts us to restart and click on yes.

Graphical user interface, text, application, email

Description automatically generated

Once the plugin is installed you can right click and then go to spotbugs and click on find bugs.

A screenshot of a computer

Description automatically generated with medium confidence

These are the bugs that are found using spotbugs

Text

Description automatically generated with medium confidence

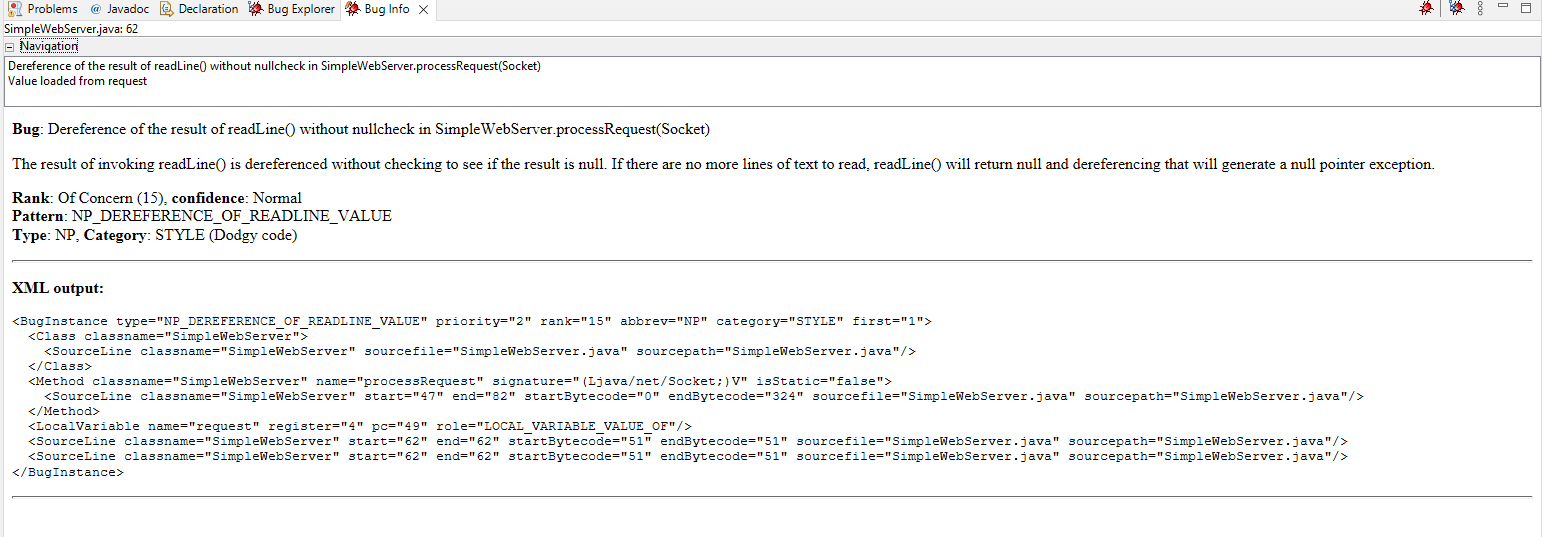
Below is the information on the bugs and where the bug exists in the program



Graphical user interface, text, application, email

Description automatically generated





For pmd using the same process install pmd eclipse plugin 4.29.0 and then restart

Graphical user interface, text, application

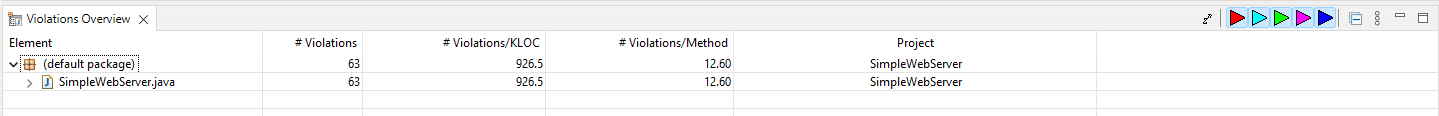
Description automatically generated

Once the pmd is installed we can rightclick and go to pmd and click on check code

A screenshot of a computer

Description automatically generated

Below is the generated report for the pmd check



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Comparison/Contrast Tools:

Does the tool analyze source or binary as input?

* PMD used for the analysis of source code. Spotbugs analyzes the binary as input

Which category of tools is it?

* Spotbugs is type checking and program verification type
* PMD is style checking and program understanding type

Show an example (if one exists) of a finding that is reported by one tool and not others.



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

We cans see that the bugs reported in spotBugs on line 63 is not there ins PMD report

Show an example (if one exists) of a finding reported by multiple tools.

Graphical user interface, text, application

Description automatically generated

The bug in lone 29 is shown in both the reports

For the known flaw in the code used, document which tools reported it (true negative) and which tools did not (false positive).

* PMD reports way more bugs when compared to Spotbugs. An example would be, PMD shows that the socket is not closed in the code which is a flaw. If the socket is bound to any of the addresses then the address is permanently lost as the socket is not closed.

Result:

Graphical user interface, text

Description automatically generated

We use the most aggressive approach that is possible

PART 2:

This is a simple code for area of a rectange

Table

Description automatically generated

The scanner classs is not closed in this program which can cause errors.

By manual analysis we close the scanner class in the next part of the task

Result:

Doing the sportbugs check we don’t find any bugs in the code and then when we use the PMD check we have found the violations which are showed in the above image

We fixed the code that is causing the violations In the before image and have cleared most of the violations that have occurred

1. We declare variables in different lines
2. The useless parentheses violation is auto fixed while clearing the other bugs

The image after fixing the bugs is shown below

Graphical user interface, application, table

Description automatically generated