Derek McPeak review and changes to

Doug Lundin’s CSC245\_Project3\_insecure

03/11/2022

Table of contents

Page 3 – Working environment & code functionality

Page 4 – Introduction/summary of findings

Page 5 – Source code and Mitigated code output

Page 6 – Issue 3

Page 7 – Issue 2

Page 8 – Issue 1

Page 9 – Issue 5 & 4

Page 10 – Issue 7

Page 11 – Summary & Conclusion

Page 12 – Appendix A – Source Code

Page 13 - Appendix B – Mitigated Code

Working environment & code functionality

Graphical user interface, text

Description automatically generated

Mr. Lundin requested me to review his project called “CSC245\_Project3\_insecure”. I had one week to review and test the code, in this week there were six separate issues that were found. In the comment of the code, each issue is labeled numerically, and the definition to each number is labeled below. \*Issue 6 was not present

|  |  |  |
| --- | --- | --- |
| **Issue** | **Authoritative Source** | **Specific vulnerability** |
| 1 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Minimize the scope of variables |
| 2 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Do not declare more than one variable per declaration |
| 3 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Use meaningful symbolic constants to represent literal values in program logic |
| 4 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Use braces for the body of an if, for, or while statement |
| 5 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Do not place a semicolon immediately following an if, for, or while condition |
| 6 | Java Coding Guidelines, 75 recommendations for reliable and secure programs | Do not attempt to help the garbage collector by setting local reference variables to null |
| 7 | The CERT Oracle Secure Coding Standard for Java | FIO14-J. Perform proper cleanup at program termination |

Demonstrating there is no meaningful variation of the output of code

Text

Description automatically generatedSource code output:

Text

Description automatically generatedMitigated code output:

Issue 3

Starting top down, the first issue to be discovered was issue 3, the variables “OWM and LOCATION” were initialized at the same time, this is a problem because it can create confusion about what the class type the variable is, and its value. This may cause it to be mishandled. These were separated to their own individual initializations.

Source:



Mitigation:



Issue 2

Regarding issue 2, the name of the variables “OWM, conn, rd” are obscure and they no real meaning to other people other than to the author, I changed the name of “OWM” to API\_KEY, “conn” to connection, and “rd” to reader” which describe the purpose of the variable more accurately. These changes will help with readability of the code by making it clearer what their use is for.

Source:





Mitigation:





Issue 1

Another vulnerability was identified which was the initialization of a variable called urlString in the Main method that was never called in its scope, while it is used in the getTempForcity method, it is not used in Main. This violates issue 1, there for, while it can provide use in the future it has since been commented out but kept because it can be used in the future. Commenting it out eliminates it as attack vector but remains dormant.

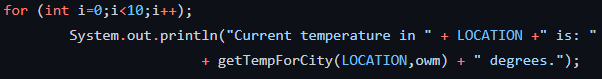
Source: 

Mitigation:

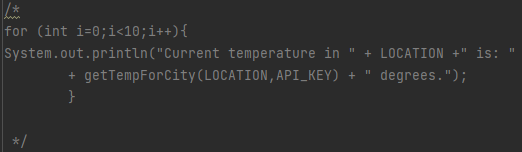


Issue 5 and 4

The next vulnerability that was found was issue 5 and 4. The first issue, number 5, the *for loop*  that was used first had a semi colon immediately following the header. This can create unexpected behavior, as doing this will cause the loop to not run properly by only running once and not ten times as intended. Issue 4 was the for loop does not contain braces around the body, I added the braces. This is a best practice to improve uniformity and intent. And lastly regarding the *for loop* and its intent, it seems to make ten calls and ten prints for the same information. Subsequently the code has been commented out but kept for future use if modifications are made to it to justify its use.

Source: 

Mitigation:

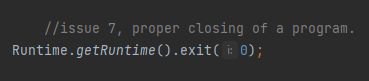


Issue 7

The last issue to be found was number 7, or a lack of proper clean-up of memory and exiting of a program.

There is no source code to provide since a program exit doesn’t currently exist.

Mitigation:

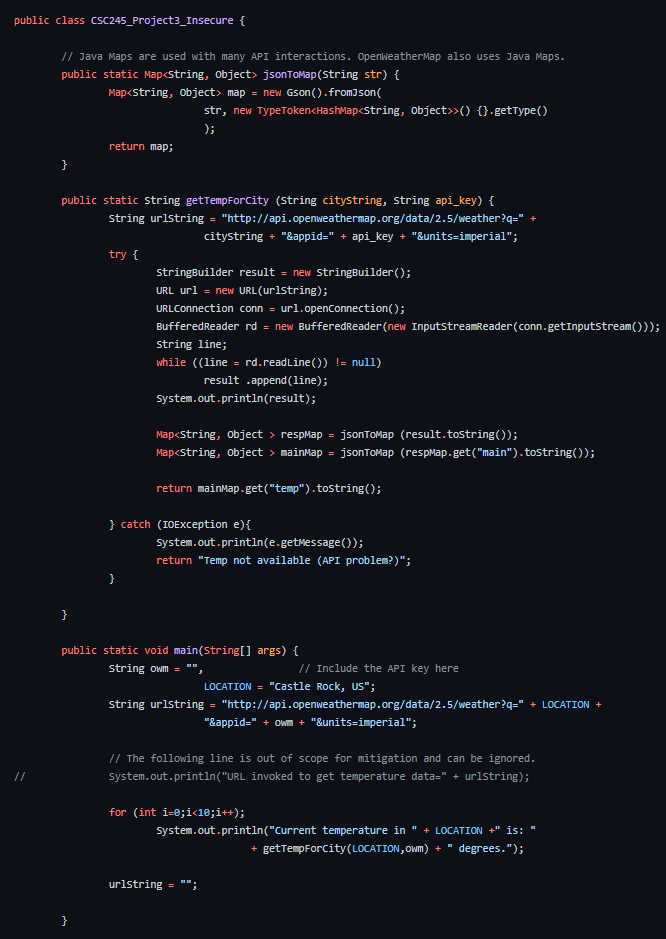


If all changes are unable to made, I have listed the proposed changes that need to be made. 1 being most the most important.

1. Issue 7, adding a proper runtime exit.
2. Issue 4 and 5, removing the semi colon that follows the header, and adding braces to the body
   1. AND/OR the deletion of the for loop
3. Issue 3 and 2, separating the initializations of “OWM” and “LOCATION”, and changing the name from “OWM” to “API\_KEY”
4. Issue 1, removing the URL String from main since it is not currently being used.

Conclusion – Best case scenario all changes are implemented, while not all changes are as vital as others, all changes should be implemented if cost allows. These offered mitigations should be used as reference for future code to ensure code integrity.

Appendix A – source code



Appendix B - mitigated code

