Derek McPeak review and changes to

Doug Lundin’s CSC245\_Project3\_insecure

03/11/2022

Table of contents

Page 3 - Introduction/summary of findings

Page 4-5 – Analysis

Page 6 – Prioritization and conclusion

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 |

Starting top down, the first issue to be discovered was issue 3 and issue 2, regarding issue 3 both “OWM and LOCATION” were initialized at the same time. Regarding issue 2, the name of the variable “OWM” is obscure and has no real meaning to other people, I changed the name to “API\_KEY” to describe the purpose of the variable more accurately. Both of these will help with readability of the code.

Source:



Mitigation:



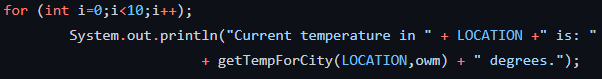
Another vulnerability was identified which was the initialization of a variable that was never called in its scope. This violates issue 1, there for, while it can provide use in the future it has since been commented out but kept because it can be used in the future. Commenting it out eliminates it as attack vector but remains dormant.

Source: 

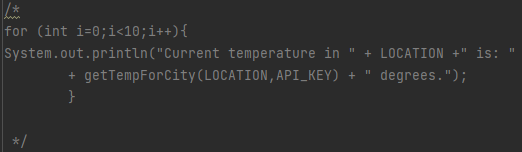
Mitigation:



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 run even if it isn’t 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: 

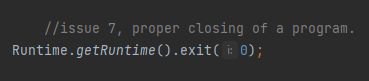
Mitigation:



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

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

Mitigation:

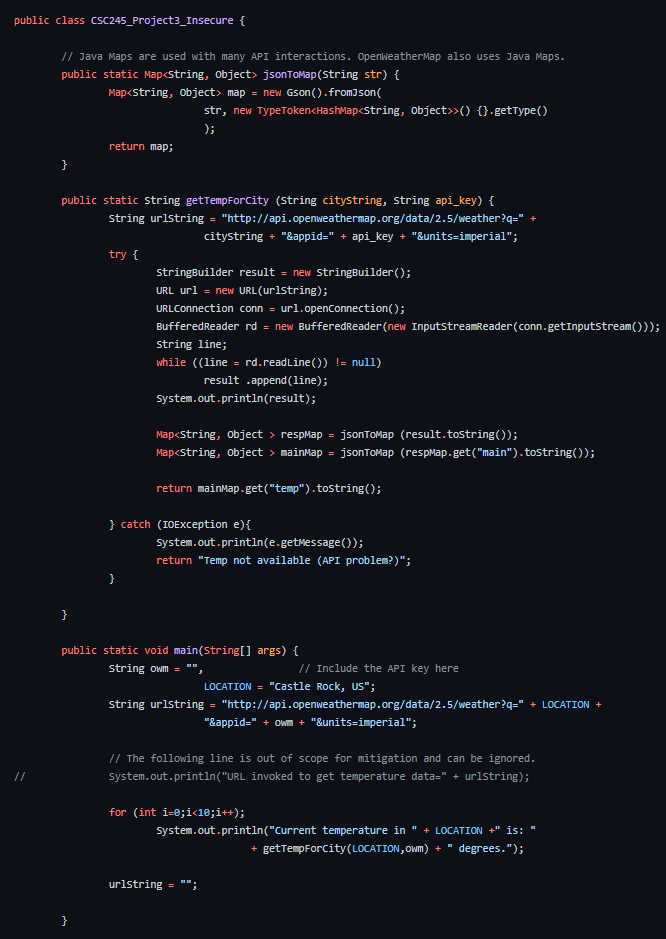


If all changes are unable to made, I have listed the proposed changes that need to be made. 1 being most the most important.

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

