Jackson State University
Department of Computer Science
CSC 438-01/539-01 Systems and Software Security, Spring 2014
Instructor: Dr. Natarajan Meghanathan

Project 4: Testing for Software Security: Source Code Analysis

Due: April 16, 2014, 7.30 PM
Max. Points: 100 (50 points for each program and their source code analysis)

Program 1: Write a Java program that reads a file (whose name is input by the user) containing a sequence of integers (one integer per line), and computes/prints the sum and average of the integers. Execute the code with a text file containing at least 5 integers and show that the Java program does compute their sum and average. Make sure the file (whose name is input by the user) is a text file and is in the same directory as that of the program; otherwise, the program should return an error.

Program 2: Write a Java program that generates a sequence of 10 random integers (between 1 to 100) and determines whether they are even or odd, as well as writes the results to a text file, one line per integer. The information on each line should be the integer value and whether it is odd or even (as shown in the sample below). The name of the file to write to should be input by the user and the file should be located/created in the same directory as that of the program; if the user attempts to input a path (i.e., the target file to write appears to be located in another directory), then the program should terminate with an error message. Also, the file to which the contents are written must be saved as a text file. Execute your code and show the contents of the text file.

Sample contents of the text file:

10	even
25	odd
7	odd
9	odd
30	even

Source Code Analysis and Recording: For each of the above two independent Java programs, your program should be analyzed with the HP Fortify Source Code Analyzer and the only vulnerabilities that need to be listed in the final scan of the program must be the poor logging practice vulnerabilities and any J2EE bad practices vulnerability that arises due to the use of String args[] as parameters for the main function or use of System.exit( ). Like in the Case Study document, show the incremental refinements that you had to do if the initial version of the Java program that you come up with (to meet the above functional requirements) is analyzed to have vulnerabilities other than the poor logging practice and J2EE bad practices vulnerabilities. Record a video showing the various stages of your refinements starting from an initial version and explain/show how you fixed the vulnerabilities. Also, clearly state which vulnerability you removed with each refinement. Do not attempt to remove more than one vulnerability in a single refinement.

Desktop Video Recording
You could try using one of the **desktop recording software** (or anything of your choice):
CamStudio: http://sourceforge.net/projects/camstudio/files/legacy/
Debut: http://www.nchsoftware.com/capture/index.html

Installation of Source Code Analyzer: Look at the video posted in JSU Blackboard on the procedure to install the Source Code Analyzer in a personal computer.