SE-4930: Developing Secure Software

Lab 7: Of Bugs and Finding Them

Due by 23:59 10/28/2014

Note: You may work in teams of your choice if you desire. However, teams should not be larger than 2 – 3 people.

Objectives

- Employ static analysis to analyze a software application for security vulnerabilities

Introduction

Programmers make mistakes. Everyone does it. We find some of those mistakes through testing, but others are missed. We find some of those mistakes through reviews, but some are missed. So what else can we do to catch the other bugs?

One technique commonly employed in the software security arena is static analysis. Integrating static-analysis tools and techniques into the development process can yield significant reductions in development testing and field failures, and from a security aspect, can inoculate an application against certain vulnerabilities.

Static code analysis is a broad term for a set of techniques used to aid in the verification of computer software without actually executing the programs. The sophistication of the analysis varies greatly depending on the tool employed. The simplest tools often only search source code for text pattern matches or calculate basic program metrics (such as Cyclomatic complexity or Halsted complexity) to determine the likelihood of problems arising from a given code segment. More advanced static-analysis tools act as an advanced compiler for the source code, deeply analyzing both execution and data flow for faults that may lead to a field failure. Some of the most advanced tools will also include link information in their analysis to determine higher level problems.

This lab serves two purposes. You will be exposed to a commercial software security static analysis tool from Fortify Software. This tool allows one to analyze advanced software projects for security vulnerabilities. Second, through the usage of this tool, you will be able to learn more about vulnerabilities which may present themselves in a secure software environment.

Step 1 – Downloading and Installing Fortify

You will need to download two files from the web site in order to install the Fortify analysis tool. The first file, an executable, is the installer for the Fortify analysis tool. This is a large file, appropriately 500MB in size. Only the 64 bit version is available.

During the installation, you will be prompted to enter the license file. This is also available for download from the web site.

During the installation, unless otherwise instructed, select the default options.

Step 2 – Download the HackMe Bookshop

Last week, you broke into the HackMe bookshop using a combination of techniques. The going to analyze the source code for this application, looking for vulnerabilities in the Java source code. The source code is written as a J2EE application.

In downloading and extracting this archive, place the code into a known directory.

Step 3 – Running Fortify

Start the Fortify Audit Workbench. Create a new java project, and select the Hackme Bookshop application. Treat the Hackme Bookshop as a Java 1.5 application. For right now, look at all security issues as well as all code quality issues, and assume that the application may run in elevated privileges.

Step 4 - Analyzing the Results

After the Fortify tool completes, go through and analyze all of the findings. How many hot issues were found? How many warnings were found?

For each bug found, use the summary tab to enter an analysis of the uncovered bug and to categorize the issue. In doing so, you may find it useful to look at the diagrams which show a sequence that may be used to attack the system using the given vulnerability.

Step 5 - Generating a Report

After the analysis has been completed, generate a Fortify Security Report for the project. Save this report in pdf format and submit to the course website

Step 6 – Running on your own code

If you have written a web application as part of your coursework here at MSOE (i.e. web applications, SDL, or Senior design) try running it through the tool and see what the results are. You are welcome to use Senior design code or SDL code if you would like to do so.

Deliverables

Each team shall submit a report in pdf format with the following information

1. *Title Page*
    a. Name of all team members,
    b. course
    c. date.
2. *Analysis of Hackme Bookstore*
    a. *If you had done a traditional peer review, how many of these issues would you have found?*
    b. *If you had done a traditional peer review, how long would it have taken and would you have found these defects?*
    c. *Did the tool seem to be valuable from a security standpoint?*
3. *Analysis of your code*
    a. What code did you analyze? (Describe briefly.)
    b. What did you find in your code that you might be concerned about?
    c. *Did the tool seem to be valuable from a security standpoint for this project?*
4. *Things gone right / Things gone wrong –*
    a. Discuss the things which went well with this lab,
    b. Discuss the problems with the lab.
5. *Conclusions*
    a. What have you learned from this experience?

This report is separate from the Fortify Security Report generated by the tool, which should also be submitted.

Note:

Now that you have the tools, it is expected that you will use them in the development of your class projects. You may not have completely clean code, but it is important to at least know where the vulnerabilities may lie and be able to explain them.

Note about Licensing:

This software is expensive, and we have obtained Academic licenses at no cost through an academic partnership. However, this license is only valid provided the tools are used for academic purposes. While you are free to use this to analyze code in other courses (SDL, senior design, for example), the use of these tools outside of MSOE is not permitted.