CSE 7359 – Software Security

Lab 5/Project – Static Code Analysis

Due: May 4

\*I am letting you choose which of the last 2 assignments you would like to use for your project submission, so everything is due May 6 for all students.

Please Note which option you have chosen: This is my Project  Final Lab

IMPORTANT: Also based on your choice submit the assignment under the proper Assignment in Black Board (Project vs. Lab 4)

**Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Objective: Perform static code analysis on a significant piece of software.

Detail: Any kind of code review requires patience, an eye for detail, and extensive knowledge of the types of problems that constitute a risk. A security audit is no different, but instead of thinking simply "What could go wrong?," the auditor must consider "What could an attacker force to go wrong?." The auditor's role is to pare down this infinite search space and identify the most dangerous problems and weaknesses in an application. Experience is invaluable, and experience comes from practice.

Download the source code for Foundstone’s Hacme Bank v.2.0 at the following website <http://www.mcafee.com/us/downloads/free-tools/hacme-bank.aspx>

Your task is to perform a manual static code analysis of the application and produce a findings report with **remediation suggestions** and **priorities** (i.e. High/Med/Low). You must find at least **10 vulnerabilities** in this application (there are many more than that). You may document up to 4 multiple instances of the same type of vulnerability (i.e. 4 SQL injections, 4 XSS, etc). You must identify the **file/line of code** of the finding, the **finding type** (i.e. XSS, Buffer Overflow, etc), **describe the vulnerability**, and give **remediation advice** for how you would fix the vulnerability (be specific). Be sure to **rate the vulnerabilities** with how critical you believe the finding to be. You can use your own criticality rating or borrow one from a pre-existing ranking system. Be sure to **include short code clips** of the lines where you found the vulnerabilities so that I don’t have to open every file to find them. Do not use an automated Static Code Analysis tool for this assignment, the point is to understand how to look for vulnerabilities, not how to use a tool to help find them for you. Tools have their limits when it comes to understanding the real-world logic of an application.

IMPORTANT NOTE: This is not just a list of vulnerabilities. Your findings report should be very professional as if you were submitting to a real-world customer. I will evaluate not only on how well you do in prioritizing vulnerabilities and remediation advice, but also on the level of professionalism.

In addition to your analysis report, create a write up on the process you used, what interesting things you learned; what pitfalls you encountered. Your report and write-up should convince me that you know the ins and outs of static code analysis. Comment on the current state of static code analysis and whether or not you feel that it should be an integral part of software development.

Here are some *suggestions* for items to consider for the analysis and write-up:

• How large is the application?

• What specific technologies are involved?

• What is the basic design of the application?

• Who are the likely attackers?

• What would an attacker hope to achieve?

• How are the developers trying to protect the application?

• What areas of the application will likely attract the attention of an attacker?

• What sorts of techniques might an attacker use to subvert the application?

• What risks would a successful attack pose to the company?

• How and when is main() executed?

• Which pieces of input to the program can an attacker control?

• Has the developer made assumptions that an attacker could violate?

• How is this code vulnerable to attack?

Submission:

Submit two documents this time. One is your Static Code Analysis Findings Report, and the other is the Write-Up on the process and methods used to generate the findings report (see descriptions above for both of these documents)