Software Security in Depth

Course Summary

Description

This course teaches the students how to develop secure software in today's complex internetworked environment. Students will receive a deep and thorough understanding of the most prevalent and dangerous security development defects in today's complex and internetworked computing environments. Additionally, they will learn practical and actionable guidance on how to avoid making these common mistakes.

The class starts with a description of the security problems faced by today's software developer, as well as a detailed description of the CWE/SANS 25 Most Dangerous Programming Errors (www.sans.org/top25errors). These defects are studied in instructor-lead sessions as well as in hands-on lab exercises in which each student learns how to actually exploit the defects. (The labs are performed in safe test environments.)

Remediation techniques and strategies are then studied for each defect. Practical guidelines on how to integrate secure development practices into the software development process are then presented and discussed.

Topics

- Preparation Phase: Understanding the problem
- Overview of available solutions
- Lab setup and demo
- Software security defects
- Processes – Design activities
- Processes – Security testing
- Processes in depth – Static code analysis
- Getting started
- Contest – The Challenge!

Audience

The ideal student for this tutorial is a hands-on web application developer or architect who is looking for a fundamental understanding of today's best practices in secure software development.

Duration

Five days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Software Security in Depth

Course Outline

I. Preparation Phase: Understanding the problem
- A. What are the issues that result in software that is susceptible to attack?
- B. Why do software developers continue to develop weak software?

II. Overview of available solutions
- A. Top-level discussion of best practices for developing secure software
- B. Security activities that can be integrated throughout a typical software development lifecycle

III. Lab setup and demo
- A. Students install and configure software tools to be used in the upcoming exercises
- B. The instructor demonstrates the tools and runs through a sample exercise to ensure all students can use the tools correctly

IV. Software security defects
- A. Introduction to CWE/SANS top 25 software security weaknesses
- B. Exploiting these weaknesses discussion
- C. Class exercises of the most common application weaknesses
- D. Preventing the weaknesses during software development
- E. Continued demonstration and class exercises of CWE/SANS top 25 software security weaknesses

V. Processes – Design activities
- A. Architectural risk analysis in detail
- B. Attack resistance
- C. Ambiguity analysis
- D. Weakness analysis
- E. Compare and contrast common processes for reviewing designs

VI. Processes – Security testing
- A. Black box vs. white box security testing of software
- B. Overview of common testing methodologies and tools
- C. Penetration testing
- D. Fuzz testing
- E. Dynamic validation

VII. Processes in depth – Static code analysis
- A. Description of static code review processes
- B. Automated vs. peer review comparison of benefits and weaknesses
- C. Background of available automated static code review tool technology
- D. Integrating a static code review tool into a software development process effectively

VIII. Getting started
- A. Key elements to succeeding with a software security initiative
- B. Developing an action plan
- C. First steps

IX. Contest – The Challenge!
- A. Students are put to the test to see who can finish the contest first
- B. The lessons taught in this class are used to solve a puzzle that is highly representative of attacking/testing a modern web application