ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Inside the Biggest Web Attacks and How to Defeat Them

Course Summary

Description

This course teaches the students how to develop secure web applications in today's complex internetworked environment. Students will receive a deep and thorough understanding of the most prevalent and dangerous security defects in today's applications. Additionally, they will learn practical and actionable guidelines on how to remediate against these common defects in Java/J2EE and how to test for them in their own applications.

This class starts with a description of the security problems faced by today's software developer, as well as a detailed description of the Open Web Application Security Project's (OWASP) Top 10 of 2010 security defects. 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 to "break into" a real web application. (The labs are performed in safe test environments.)

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

Topics

- Preparation Phase: Understanding the problem
- Overview of available solutions
- Lab setup and demo
- Exploiting web application weaknesses
- Secure development processes
- Introduction to design review exercise
- Processes in depth – Threat Modeling
- Threat modeling exercise
- Processes in depth – Static code analysis
- Static code analysis exercise
- Processes in depth – Security testing
- A day in the life of a web application

Audience

The ideal student for this course is a hands-on web application developer, architect, or information security practitioner with software experience. The course establishes a solid fundamental understanding of today's best practices in secure software development.

Prerequisites

Students should have some software experience.

Duration

Three days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Inside the Biggest Web Attacks and How to Defeat Them

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
- C. Review of web application basics
  1. HTTP methods (e.g., GET, POST)
  2. Identification and authentication
  3. Session management

IV. Exploiting web application weaknesses
- A. Introduction to OWASP top 10 (and other) security weaknesses in web applications
- B. How do attackers exploit these weaknesses?
- C. Class exercises of the most common web application weaknesses

V. Secure development processes
- A. A detailed look at three common secure development methodologies, and their strengths and weaknesses
  1. Microsoft's SDL
  2. Cigital's Touchpoints
  3. OWASP's CLASP
  4. Group discussion of the feasibility of the processes

VI. Introduction to design review exercise
1. Group exercise to review an example of a flawed design for security weaknesses

VII. Processes in depth – Threat Modeling
- A. Reviewing designs using threat modeling
- B. Finding the weaknesses in an application architecture
- C. Documenting how the weaknesses can be exploited
- D. Deciding what and how to mitigate the weaknesses

VIII. Threat modeling exercise
- A. Hands-on team threat modeling exercise
- B. Review a design step by step for weaknesses
- C. Discuss what should be mitigated and how

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

X. Static code analysis exercise
- A. Group exercise in which a simple program is analyzed using a commercial static code analysis tool
- B. The results are reviewed and analyzed by the class
- C. Group discussion about how to best utilize a static analysis tool

XI. Processes in depth – 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

A day in the life of a web application
- A. Review a simple, flawed web application
- B. Add fixes against common web app flaws
- C. Integrate enterprise security features into the application