Fundamentals of Software Security

- Understand how to build and test secure software
- Practice identifying software vulnerabilities within code
- Get techniques to start implementing a security improvement program

Software security is the weakest link in information security today. It is common for software applications to contain security vulnerabilities that allow unauthorized personnel to compromise systems, steal intellectual property, or disclosure sensitive customer data. To combat these risks, a proactive approach to building secure software applications is necessary.

This two-day course teaches the fundamentals of software security – providing participants with a comprehensive understanding of how to build and test secure software. In this course you will learn:

- why software is insecure and how to best address these concerns
- how to build security into applications from the ground up
- where to integrate security testing into a software development process
- how continuous integration can be leverage to automate security analysis
- what secure software development approaches are available for use
- how to measure the maturity of your software security approach
- how to combat malicious code

Fundamentals of Software Security includes exercises to practice identifying actual software vulnerabilities within code and learn how to avoid introducing them. Tools and techniques for web application security testing, secure code scanning, and fuzz testing are discussed and applied to software.

Attendees will leave this class with an in-depth understanding of how to build security into software from the ground up as well as analyze software to identify existing risks and vulnerabilities.

Who Should Attend
The audience includes software developers, software architects, and software designers. A strong educational and experiential background in software development is recommended.

Course Outline

Introduction to Software Security
History of information security
The software security problem
How attackers think
Approaches to solving the problem
Roles in software security
Discussion: Understanding your software

Common Software Security Attacks
Web application attacks
- XSS
- CSRF
- SQL Injection
- Session Hijacking
- Command Injection

Security Assurance
Threat modeling/Architectural risk analysis
Case study: Developing threat models
Secure code review
Case study: Reviewing code
Penetration testing / red teaming

Secure Software Development Approaches
Microsoft SDL
Security Touchpoints
Secure Agile
- Security stories
- Secure TDD
- Pair programming
- Secure CI

Case study: Security testing

• XML Injection
Demos: Example web app attacks
Traditional application attacks
• Buffer overflows
• Race conditions
Discussion: Where are your security problems?

Building Secure Software
Secure requirements
Secure architectures and defensive design
Securing coding practices
Security testing
Case study: Building security requirements

Getting Started
Assess your risks
• Threat models
• Code scanning
• Security testing
Fix critical vulnerabilities
Move toward building security in

Wrap up
References
Q & A