

## Building and Testing Secure Web Applications Course Outline

### Aspect Security, Inc.

Contact:	Peter Dean
Email:	<a href="mailto:peter.dean@aspectsecurity.com">peter.dean@aspectsecurity.com</a>
Phone:	(973) 668-5595
Address:	9175 Guilford Road, Suite 300 Columbia MD, 21046

#### RESTRICTION NOTICE

THIS DOCUMENT CONTAINS MATERIAL THAT IS PROPRIETARY TO ASPECT SECURITY.  
INFORMATION CONTAINED HEREIN SHALL NOT BE DISTRIBUTED FOR ANY REASON WITHOUT WRITTEN PERMISSION.

## 1 Executive Summary

---

Training developers and software testers in application security offers one of highest returns on investment of any security investment by eliminating vulnerabilities at the source. Aspect's Building and Testing Secure Web Applications training raises developer awareness of application security issues and provides examples of 'what to do' and 'what not to do.' The class is lead by an experienced application security practitioner and is delivered in a very interactive manner.

This class includes hands-on exercises where the students get to perform security analysis and testing on a live web application. This specially designed environment includes deliberate flaws the students have to find and diagnose. Students gain hands-on experience using freely available web application security test tools to find and diagnose flaws and learn to avoid them in their own code.

## 2 Audience

---

The intended audience for this course is:

- Software developers in any web environment
- Software testers
- Security specialists
- Application architects

## 3 Learning Objectives

---

At the highest level, the objective for this course is to ensure that developers are capable of designing, building, and testing secure applications and understand why this is important.

Topic	Learning Objective
<b>HTTP Fundamentals</b>	Understand and be able to employ the security features involved with using HTTP (e.g., headers, cookies, SSL)
<b>Design Principles and Patterns</b>	Understand and be able to apply application security design principles.
<b>Threats</b>	Be able to identify and explain common web application security threats (e.g. , cross-site scripting, SQL injection, denial of service attacks, "Man-in-the-middle" attacks, etc.) and implement mitigation techniques.
<b>Authentication and Session Management</b>	Be able to handle credentials securely while providing the full range of authentication support functions, including login, change password, forgot password, remember password, logout, reauthentication, and timeouts.
<b>Access Control</b>	Be able to implement access control rules for the user interface, business logic, and data layers.
<b>Cross Site Request Forgery (CSRF)</b>	Be able to develop defenses against Cross Site Request Forgery attacks.
<b>Input Validation</b>	Be able to recognize potential input validation issues, particularly injection and Cross-site Scripting (XSS) problems, and implement appropriate input validation mechanisms for user input and other sources of input.
<b>Command Injection</b>	Understand the dangers of command injection and techniques for avoiding the introduction of this type vulnerability.

<b>Error Handling</b>	Be able to implement a consistent error (exception) handling and logging approach for an entire web application.
<b>Cryptography</b>	Learn when to apply cryptographic techniques and be able to choose algorithms and use encryption/decryption and hash functions securely.
<b>Auditing and Logging</b>	Be able to select and implement appropriate auditing/logging capabilities.
<b>Denial of Service</b>	Understand the variety of denial of service attacks and the techniques that can be employed to reduce the likelihood of a successful denial of service attack.
<b>Verification</b>	Be able to review their applications for common security vulnerabilities using code review and penetration testing techniques.

## 4 About the Instructors

---

Aspect Security has been working with development teams around the country for years to help them identify, diagnose, and address security issues throughout the application development lifecycle. Through these efforts, they have learned the key practices that development and project managers, and key support personnel must know to achieve secure applications.

Aspect's instructors are full-time application security specialists that spend the majority of their time working with clients to secure the nation's most critical applications. Leveraging this practical experience brings the class to life. Students will gain valuable insight into lessons learned from other development organizations. Our instructors also make themselves available to you for application security questions after the course is complete.

Aspect is a Founding OWASP Member and supports several OWASP projects. In particular, Aspect conceived the OWASP Top Ten project and led the effort to build the document. We also built WebGoat and Stinger and donated them to the OWASP effort. Aspect personnel assist with the management of the OWASP Foundation and help run the OWASP AppSec conference series.

## 5 Agenda

---

Day 1		Day 2	
8:30	Welcome and Introductions	8:30	Welcome to Day 2
8:40	Understanding Web Application Security	8:40	How to Use Databases Securely ▲
9:00		9:00	
9:20	Understanding HTTP ▲	9:20	Error Handling and Logging ▲
9:40		9:40	
10:00	BREAK	10:00	BREAK
10:20	How to Authenticate Users ▲	10:20	BREAK
10:40		10:40	
11:00	How to Manage User Sessions ▲	11:00	How to Ensure Availability ▲
11:20		11:20	
11:40	How to Manage User Sessions ▲	11:40	How to Ensure Quality ▲
12:00		12:00	
12:20	LUNCH	12:20	LUNCH
12:40		12:40	
13:00	How to Control Access ▲	13:00	How to Access Services Securely ▲
13:20		13:20	
13:40	How to Control Access ▲	13:40	Setting Security Policy
14:00		14:00	
14:20	BREAK	14:20	BREAK
14:40	How to Protect Against XSS ▲	14:40	Integrating Security into the SDLC
15:00		15:00	
15:20	How to Architect Input Validation ▲	15:20	Integrating Security into the SDLC
15:40		15:40	
16:00	How to Protect Sensitive Data ▲	15:40	Challenge ▲
16:20	Wrap-Up Day 1	16:00	Wrap-Up Day 2
16:30		16:20	
		16:30	

▲ Hands-on Testing Exercise

## 6 Outline

---

### 1) Introduction

Section Overview: This section describes and introduces the course, and instructors. It also provides setup instructions for the course exercises.

- a) Training Program Introduction
- b) Course Objectives, Approach, and Layout
- c) Students Introduce Themselves
- d) Intro to Aspect Security/Instructors
- e) Discussion of Applicable Corporate Initiatives
- f) Review of Course Agenda
- g) Install and Setup Testing Environment

## 2) Understanding Web Application Security

**Section Overview:** This section introduces what web application security is and focuses on how vulnerable software exposes a company's assets. It compares and contrasts application security with network and host security. It also briefly introduces the concept of risk. Finally, it describes the current state of the application security market along with different forces involved with its evolution.

- a) Introduction to What Application Security Involves
- b) Differences between Application and Network/Host Security
- c) Understanding the Application Security Problem
- d) Test Your Hacker IQ
- e) Thinking about Risk
- f) OWASP
- g) Market Forces and Trends

## 3) Understanding HTTP and Web Technologies

**Section Overview:** This section is intended to provide the foundations needed to understand the upcoming application security concepts. It begins by describing the HTTP protocol and how it relates to web applications. It dives into various aspects of the protocol, in detail, to assist in the understanding of the entire communication path from client request, server processing, server response, and browser interpretation. It then discusses how a hacker proxy can be used to modify HTTP requests and where this proxy fits into the big picture. Finally, we begin the first hands-on lesson which is intended to get the students familiar with the hands-on application and comfortable using a proxy.

- a) HTTP Protocol (Requests, Responses, Headers, Cookies, Parameters, Response Codes)
  - i) Security of GET vs. POST
  - ii) SSL and Certificates
  - iii) Redirect and Forward
- b) Introducing a Security Testing Proxy
  - i) WebScarab Overview
  - ii) Example Lab Description
- c) Exercises and Labs
  - i) Hands On Testing Exercise: WebGoat HTTP Basics
  - ii) Hands On Testing Exercise: WebGoat and Proxy

## 4) How to Authenticate Users

**Section Overview:** This section introduces common web authentication methods along with their strengths and weaknesses. It discusses best practices associated with authentication and uses hands-on lessons to demonstrate some common authentication mistakes. Through this we discuss different technology specific authentication uses and configurations.

- a) Overview
- b) Authentication Mechanisms
- c) Common Authentication Approaches (LDAP, Database)
- d) How to Protect Credentials from Disclosure
- e) How to Protect Against Brute Force Attacks
- f) How to Provide Password Management Functions
- g) How to Protect Against Phishing
- h) Exercises and Labs
  - i) Hands On Testing Exercise: WebGoat – Basic Authentication
  - ii) Hands On Testing Exercise: WebGoat – Authentication Cookies
  - iii) Spot the Bug(s): Flawed Password Change Page

## 5) How to Manage User Sessions

**Section Overview:** This section introduces what session management is and how it works within a web application environment. It discusses common mistakes developers make regarding session management and attacks that stem from these mistakes. The section discusses best practices associated with session management and technology specific implementation approaches.

- a) Introduction to HTTP Sessions
- b) Explanation of Session Lifecycle (login, logout, reauthentication, timeouts)
- c) How to Protect Against Session Hijacking
- d) Exercises and Labs
  - i) Hands On Testing Exercise: WebGoat – Weak Session Identifier
  - ii) Spot the Bug(s): Logout Flaws

## 6) How to Control Access

**Section Overview:** This section introduces access control in a web environment and the various complexities associated with implementing strong access protections. It walks through the importance of checking all access to sensitive functionality, defining application roles and functions, not relying only on presentation rendering, and implementing access controls at different level, including: declarative (URL), programmatic (API) and instance (data) level. Throughout the section various technology specific access control uses are discussed and demonstrated. This section also includes common best practices associated with access control.

- a) Overview
- b) Defining & Architecting Your Access Control Policy
  - i) Authorization Primitives
  - ii) Defining an Access Control Matrix
- c) Presentation Layer Access Control
  - i) Single Role vs. Multi-Role Views
- d) Environment Enforced Access Control
  - i) Attack Surface
  - ii) Single Role vs. Multi-Role URLs
  - iii) Declarative Authorization
- e) Business Layer Access Control
  - i) Programmatic Authorization
  - ii) Single Role vs. Multi-Role Business Functions
- f) Data Layer Access Control
  - i) The Object Reference Problem
- g) Other Common Access Control Problems
- h) Exercises and Labs
  - i) Hands On Testing Exercise: WebGoat – Access Control

## 7) How to Prevent Cross Site Request Forgery (CSRF) Attacks

**Section Overview:** The section introduces a very common but relatively new web application attack known as Cross Site Request Forgery. It explains how and why this attack works and the consequences of such attacks. It discusses the significance of these types of flaws and then presents several approaches for how developers can defend their applications against this type of attack.

- a) Overview of CSRF
  - i) What is CSRF
  - ii) CSRF Vulnerability Pattern
- b) How to Identify CSRF Flaws
  - i) Other Names for CSRF Flaws
  - ii) Several Real World Examples
- c) How to Protect Against CSRF

- i) Misconceptions – Defenses That Don't Work
  - ii) Recommended CSRF Defenses
  - iii) Java and .NET Specific Defenses
- 8) How to Protect Against Cross Site Scripting (XSS)

Section Overview: The section introduces a very common web application attack known as Cross Site Scripting (XSS). It explains how and why this attack works and the consequences of such attacks. It will introduce and explain two types of XSS attacks (reflected and stored), demonstrate an attack, walk through various buggy code examples, and finally allow the students to apply what they have learned by executing XSS attacks using hands-on lessons. Throughout the section different technology specific protections, including validation and output encoding, are explored and discussed.

- a) Overview of XSS
    - i) Types of XSS (Stored and Reflected)
    - ii) Tricking the Browser Sandbox
    - iii) Consequences of XSS
  - b) How to Solve XSS Problems
    - i) Filters
    - ii) Input Validation
    - iii) HTTPOnly
    - iv) HTML Entity Encoding
  - c) Exercises and Labs
    - i) Hands On Testing Exercise: WebGoat – Stored and Reflected XSS
- 9) How to Architect Input Validation Solutions

Section Overview: The section provides a basis for understanding the need for input validation. It walks through common design and implementation approaches to validate user input and discusses the strengths and weaknesses associated with each approach. It will start off with the focus on threats associated with unvalidated user input. It introduces and explains these threats, demonstrates the attacks, and allows students to apply what they have learned by using hands-on lessons. Throughout the section different technology specific protections are explored and discussed as well as the best practices associated with quality attributes of proper input validation.

- a) General Input Validation Approaches
  - i) Hidden Fields
  - ii) Hands On Testing Exercise: WebGoat – Hidden Fields
  - iii) Positive Validation
  - iv) HTML Entity Encoding
  - v) Hands On Testing Exercise: WebGoat – Encoding
  - vi) Canonicalization
- b) How to Validate Outside Applications
- c) How to Validate Within Applications
- d) How to Respond to Input Validation Issues in Applications
- e) How to Validate Data from Other Sources
- f) Input Validation Checklist Questions
- g) Exercises and Labs:
  - i) Spot the Bug(s): Input Validation Flaws
  - ii) Hands On Testing Exercise: WebGoat – JavaScript

## 10) How to Protect Sensitive Data

**Section Overview:** This section discusses common cryptographic problems associated with web applications. It will demystify and dispel the myth that crypto is extremely complex to use by walking through various simple and straightforward code examples. These code examples are technology specific and include examples of encrypting, decrypting, hashing, and the use of SSL. It also discusses other common flaws that can lead to the exposure of sensitive data.

- a) Overview
- b) How to Choose the Right Algorithm
- c) How to Encrypt, Decrypt, Sign, and Hash
- d) How to Avoid Replay Attacks
- e) How to Use SSL Sockets
- f) How to Protect Sensitive Data in Caches in Applications
- g) Exercises and Labs:
  - i) Spot the Bug(s): Flawed Use of Cryptography

## 11) How to Use Databases Securely

**Section Overview:** The section provides the material necessary to use a database securely. Threats related to securely connecting to a database, validating input, using SQL, handling errors and logging, and validating results are covered. Some architectural concerns are also discussed in terms of centralizing the security functions related to accessing a database securely.

- a) Overview/Goals
- b) How to Prevent SQL Injection
- c) Protecting Database Connection Strings (usernames/passwords)
- d) How to Limit Access to Database Information using
- e) How to Use Transactions (ACID)
- f) How to Handle SQL Exceptions and Verify Results
- g) Architectural Patterns for Database Security (DAO)
- h) Exercises and Labs:
  - i) Hands On Testing Exercise: WebGoat – SQL Injection

## 12) How to Handle Errors and Log Security Events

**Section Overview:** This section introduces the importance of proper error handling and security logging mechanisms for security critical events. Throughout the section technology specific logging APIs and error handling strategies will be introduced and discussed.

- a) Overview
- b) How to Configure Error Handling
- c) Error Handling Best Practices and Danger Signs
- d) What Events to Log and What Data to Capture
- e) Logging Best Practices and Danger Signs
- f) Exercise and Labs:
  - i) Spot the Bug(s): Improper Error Handling
  - ii) Hands On Testing Exercise: WebGoat – Fail Open Authentication Pattern
  - iii) Group Exercise: What to Log?

## 13) How to Ensure Availability

**Section Overview:** This section discusses protecting applications against denial of service attacks. In particular, flooding and lockout attacks and common countermeasures are discussed..

- a) Overview of Availability

- b) Flooding (bandwidth, file system)
- c) Account Lockout (accounts, pools)
- d) Approaches to Flooding Attacks
- e) Approaches to Lockout Attacks

#### 14) How to Prevent Quality Issues from Introducing Vulnerabilities

Section Overview: Security is directly related to quality, and software vulnerabilities increase directly with the quality of the code. This section explores the importance of establishing and following a coding guideline that is tailored to address security approaches adopted by the project. Some specific code quality problems that are frequently linked with security are included. The section includes a discussion of what can be enforced automatically and some of the relevant tools.

- a) Why Code Quality and Deployment Issues Lead to Vulnerabilities
- b) General Code Quality Best Practices for Security
- c) Handling Debug and Test Code
- d) Search Engine (Google) Hacking
- e) How to Avoid Concurrency Vulnerabilities in Applications
- f) Exercises
  - i) Hands On Testing Exercise: WebGoat – Clues in HTML
  - ii) Spot the Bug(s): Find the Concurrency Flaws

#### 15) How to Access Services Securely

Section Overview: This section discusses security issues associated with external connections, and walks through various best practices. This section is used as a review of all the practices covered so far. Students should realize that all the practices they've learned for protection of a web application should apply to an external connection as well.

- a) A Pattern for Using Services Securely
- b) Applying the Pattern to Prevent Command Injection
- c) Architecting Secure Service Access
- d) Examples of How to Access Services Securely
- e) Exercises and Labs:
  - i) Hands On Testing Exercise: WebGoat – Command Injection
  - ii) Hands On Testing Exercise: WebGoat – E-Mail Exploitation

#### 16) Setting Security Policy for Security Critical Areas

Section Overview: This section briefly discusses the importance of documenting your application security policy/security requirements. It recommends that you take the best practices presented in this course and use them as a basis for establishing your security policy.

- a) Typical Likelihood and Severity of Security Issues Covered in this Course
- b) What is an Application Security Policy?
- c) Example Application Security Policy Statements
- d) Exercise:
  - i) Write a Security Policy

#### 17) Integrating Security into the SDLC

Section Overview: This section describes the importance of integrating security activities and processes throughout the entire software development lifecycle. It walks through each lifecycle stage and suggests, at a high level, security activities, methodologies and tools that when integrated into the SDLC improve the overall security posture of an organization's software.

- a) How to Figure out How Much Security You Need

- b) Building Security Throughout the Lifecycle
    - i) Proposal, Planning, Requirements, Design, Development, Test, Deploy)
  - c) Threat Modeling Overview
  - d) Security Design Reviews
  - e) Server Configuration
  - f) How to Perform Application Security Testing and Analysis
    - i) Security Tools
    - ii) Vulnerability Scanning Tools
    - iii) Penetration Testing Tools
    - iv) Static Analysis Tools
    - v) Code Review
    - vi) Reporting Tools
  - g) Exercises and Labs:
    - i) Group Exercise: Develop a Security Test Plan
- 18) References
- a) Books
  - b) OWASP Resources
  - c) Microsoft Application Security Resources
  - d) Web Application Security Consortium Guidelines
- 19) The Challenge

Section Overview: The challenge section is intended to allow the students to step back, look at what they have learned throughout the course and apply this knowledge to performing a final hack on the hands-on Challenge lesson. This lesson combines many of the vulnerabilities previously discussed into a single lesson (with multiple stages). This lesson doesn't contain any course hints, as do previous lessons (hints are included in previous lessons to guide the student through each stage of an attack). The instructor is there to assist students, but ideally this is the time to allow the students to use their creativity and the knowledge they have gained from the course to successfully compromise the final lesson.

- a) Exercises and Labs:
  - i) Hands On Testing Exercise: WebGoat – Challenge Stage 1 – Break Authentication
  - ii) Hands On Testing Exercise: WebGoat – Challenge Stage 2 – Steal the Credit Cards
  - iii) Hands On Testing Exercise: WebGoat – Challenge Stage 3 – Deface the Web Site