SHI TT8810 Application Security and Development 5 Days Instructor LED User Guide

SHI TT8810 Application Security and Development 5 Days Instructor LED User Guide

Contents

Course Outline

STIG Security | Application Security and Development (DISA STIG Training) Course TT8810: 5 days Instructor Led

About this course
DISA's Application Security and Development STIG, in conjunction with the associated checklist, provides a comprehensive listing of requirements and needs for improving and maintaining the security of software applications
and systems within the Department of Defense. This course fills in the context, background, and best practices for

fulfilling those requirements and needs. As with all of our courses, we maintain tight synchronization between the latest DISA releases and our materials. A key component to our coverage of DISA's Security Technical Implementation Guides (STIGS), this course is a companion course with several developer-oriented courses and seminars.

Application Security and Development (STIG) is a lab-intensive, hands-on application security training course essential for developers, designers, architects, QA, Testing, and other personnel who need to deliver secure applications within the DOD. In addition to teaching basic programming skills, this course digs deep into sound processes and practices that apply to the entire software development lifecycle.

In this course, students thoroughly examine best practices for defensively coding web applications, including XML processing, rich interfaces, and both RESTful and SOAP-based web services. Students will repeatedly attack and then defend various assets associated with fully functional web applications and web services. This hands-on approach drives home the mechanics of how to secure web applications in the most practical of terms.

Audience profile

This is an intermediate -level programming course, designed for experienced Java developers who wish to get up and running on developing well defended software applications using the STIG guidelines. Familiarity with Java and JEE is required, and real-world programming experience is highly recommended. Ideally students should have approximately 6 months to a year of Java working knowledge.

Take Before: Students should have basic development skills and a working knowledge in the following topics, or attend these courses as a pre-requisite:

• TT5102 JEE Web Application Development Essentials

At course completion

After completing this course, students will be able to:

- Understand potential sources for untrusted data.
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections.
- To test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses.
- Prevent and defend the many potential vulnerabilities associated with untrusted data.
- Understand the vulnerabilities of associated with authentication and authorization.
- To detect, attack, and implement defenses for authentication and authorization functionality and services.
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks.
- To detect, attack, and implement defenses against XSS and Injection attacks.
- Understand the concepts and terminology behind defensive, secure and coding.
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets.
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in Java-based web applications.
- Design and develop strong, robust authentication and authorization implementations within the context of JEE.
  • Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within

the web services arena.

• To detect, attack, and implement defenses for both RESTful and SOAP-based web services and functionality.

• Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure.

- Understand and implement the processes and measures associated with the Secure Software Development (SSD)
- Acquire the skills, tools, and best practices for design and code reviews as well as testing initiatives.
- Understand the basics of security testing and planning.
- Work through a comprehensive testing plan for recognized vulnerabilities and weaknesses

Course Outline

Session: Securing Applications Foundation
Lesson: DISA's Security Technical Implementation Guides (STIGS)

- • Purpose
- • Process

  • Areas Covered

  • Checklists

  • Scripts (SRRs)

  • Resources

  Lesson: Removing Bugs

  • Open Web Application Security Project (OWASP)

  • OWASP Top Ten Overview

  • Web Application Security Consortium

  • CERT Secure Coding Standards

  • Bug Hunting Mistakes to Avoid

  • Tools and Resource

  Lesson: Principles of Information Security

  • Security Is a Lifecycle Issue

  • Minimize Attack Surface Area

  • Layers of Defense: Tenacious D

  • Compartmentalize

  • Consider All Application States

  • Do NOT Trust the Untrusted

  • Tutorial: Working with

  • Lab: Case Study Setup and Review

  Session: Bug Stomping 101
- Purpose
- Process
- Areas Covered
- Checklists
- Scripts (SRRs)
- Resources

Lesson: Removing Bugs

- Open Web Application Security Project (OWASP)
- OWASP Top Ten Overview
- Web Application Security Consortium
- CERT Secure Coding Standards
- Bug Hunting Mistakes to Avoid
- Tools and Resource

Lesson: Principles of Information Security

- Security Is a Lifecycle Issue
- Minimize Attack Surface Area
- Layers of Defense: Tenacious D
- Compartmentalize
- Consider All Application States
- Do NOT Trust the Untrusted
- Tutorial: Working with
- Lab: Case Study Setup and Review

Session: Bug Stomping 101
Lesson: Unvalidated Data

- Buffer Overflows
- Integer Arithmetic Vulnerabilities
- Unvalidated Data: Crossing Trust Boundaries
- Defending Trust Boundaries
- Whitelisting vs Blacklisting
- Lab: Defending Trust Boundaries

Lesson: A1: Injection

- Injection Flaws
- SQL Injection Attacks Evolve
- Drill Down on Stored Procedures
- Other Forms of Injection
- Minimizing Injection Flaws
- Lab: Defending Against SQL Injection

Lesson: A2: Broken Authentication

- Quality and Protection of Authentication Data
- Handling Passwords on Server Side
- SessionID Risk Reduction

- HttpOnly and Security Headers
- Lab: Defending Authentication

Lesson: A3: Sensitive Data Exposure

- Protecting Data Can Mitigate Impact
- In-Memory Data Handling
- Secure Pipes
- Failures in TLS/SSL Framework
- Lab: Defending Sensitive Data

Lesson: A4: XML External Entities (XXE)

- XML Parser Coercion
- XML Attacks: Structure
- XML Attacks: Injection
- Safe XML Processing
- Lab: Safe XML Processing
- Lab: Dynamic Loading Using XSLT (Optional)

Lesson: A5: Broken Access Control

- Access Control Issues
- Excessive Privileges
- Insufficient Flow Control
- Unprotected URL/Resource Access
- Examples of Shabby Access Control
- Sessions and Session Management
- Lab: Unsafe Direct Object References
- Lab: Spotlight on Verizon Exploit
- Session: Bug Stomping 102

Lesson: A6: Security Misconfiguration

- System Hardening: IA Mitigation
- Application Whitelisting
- Least Privileges
- Anti-Exploitation
- Secure Baseline

Lesson: A7: Cross Site Scripting (XSS)

- XSS Patterns
- Persistent XSS

- Reflective XSS
- DOM-based XSS
- Best Practices for Untrusted Data
- Lab: Defending Against XSS

Lesson: A8/9: Deserialization/Vulnerable Components

- Deserialization Issues
- Identifying Serialization and Deserializations
- Vulnerable Components
- Software Inventory
- Managing Updates
- Lab: Spotlight on Equifax Exploit

Lesson: A10: Insufficient Logging and Monitoring

- Fingerprinting a Web Site
- Error-Handling Issues
- Logging In Support of Forensics
- Solving DLP Challenges
- Lab: Error Handling

Lesson: Spoofing, CSRF, and Redirects

- Name Resolution Vulnerabilities
- Fake Certs and Mobile Apps
- Targeted Spoofing Attacks
- Cross Site Request Forgeries (CSRF)
- CSRF Defenses
- Lab: Cross-Site Request Forgeries

Session: Moving Forward

Lesson: What Next?

- Common Vulnerabilities and Exposures
- CWE/SANS Top 25 Most Dangerous SW Errors
- Strength Training: Project Teams/Developers
- Strength Training: IT Organizations
- Leveraging Common AppSec Practices and Controls
- Lab: Recent Incidents
- Lab: Spotlight on Capitol One Exploit
- Session: Secure Development Lifecycle (SDL)

Lesson: SDL Process Overview

- Revisiting Attack/Defense Basics
- Types of Security Controls
- Attack Phases: Offensive Actions and Defensive Controls
- Secure Software Development Processes
- Shifting Left
- Actionable Items Moving Forward
- Session: Taking Action Now

Lesson: Application Security and Development Checklists

- Checklist Overview, Conventions, and Best Practices
- Generic Application Checks and Procedures
- .Net Framework Checks and Procedures (Optional)
- Web/Java Checks and Procedures (Optional)

Lesson: Asset Analysis

- Targets: Data/Entity Assets
- Targets: Functional/Service Assets
- Classifying Based on Value and Risk Escalation
- Asset Inventory and Analysis
- Lab: Case Study Asset Analysis

Lesson: Design Review

- Asset Inventory and Design
- Assets, Dataflows, and Trust Boundaries
- Risk Escalators in Designs
- Risk Mitigation Options
- Lab: Reviewing Design Features
- Secure Code Review
- Secure Code Review Process
  Organizational Aspects
- Technical Aspects
- Best Practices
- Code Crawling
- Lab: Secure Code Review

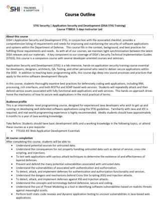
Lesson: Making Application Security Real

- Cost of Continually Reinventing
- Paralysis by Analysis

- Actional Application Security

Additional Tools for the Toolbox

Documents / Resources

	SHI TT8810 Application Security and Development 5 Days Instructor LED [pdf] User Guide TT8810 Application Security and Development 5 Days Instructor LED, TT8810, Application Security and Development 5 Days Instructor LED, Security and Development 5 Days Instructor LED, Development 5 Days Instructor LED, Instructor LED, LED

References

- User Manual