

# Java Secure Coding Camp | Attacking and Securing Java Web Applications - TT8320-J

Designed specifically for experienced Java web developers, our Java Secure Coding Camp | Attacking and Securing Java Web Applications is an immersive, hands-on training program that delves deep into the world of bug hunting, ethical hacking, and web application security.

## What You'll Learn

### Overview

Discover the cutting-edge of cybersecurity and elevate your skills as a Java Web developer with our comprehensive Bug Hunting and Application Security course. Designed specifically for experienced Java web developers, our Java Secure Coding Camp | Attacking and Securing Java Web Applications is an immersive, hands-on training program that delves deep into the world of bug hunting, ethical hacking, and web application security. Through real-world case studies, engaging labs, and expert instruction, you'll gain the knowledge and skills needed to fortify your applications, stay ahead of emerging threats, and protect your organization from costly security breaches. Upon completing this course, you will not only acquire a profound understanding of application security concepts and best practices but also enhance your problem-solving, debugging, and overall software development prowess. Empowered with these new skills, you'll be well-prepared to identify, address, and prevent security threats in your Java Web applications, ensuring a robust and secure digital environment for your organization.

**NOTE:** PCI Compliant Developer Training: This secure coding training addresses common coding vulnerabilities in software development processes. This training is used by one of the principal participants in the PCI DSS. Having passed multiple PCI audits, this course has been shown to meet the PCI requirements. The specifications of those training requirements are detailed in 6.5.1 through 6.5.7 on pages 60 through 65 of the PCI DSS Requirements 3.2.1 document.

## Objectives

With a strong focus on real-world case studies and labs, this course will sharpen your ability to identify, analyze, and resolve security issues in their applications. Working in a lab-intensive, hands-on coding environment you will:

- Master the fundamentals of secure coding and understand the stages of an exploit, focusing on defensive techniques.
- Establish foundational axioms for analyzing and addressing security in web applications, guiding your approach through this course and future endeavors.
- Learn responsible ethical hacking methods, including defect detection, bug reporting, and ensuring all activities are executed in a safe environment.
- Recognize and sidestep frequent pitfalls in vulnerability testing and bug hunting, leveraging best practices.
- Gain insight into the significance of multilayered defense strategies, evaluating the effectiveness of layered defenses through hands-on testing.
- Identify and handle untrusted data sources, understanding the associated risks like denial of service, cross-site scripting, and injections.
- Dive deep into authentication and authorization, pinpointing vulnerabilities and learning how to fortify these crucial security areas.
- Understand and counteract web-specific threats such as Cross-Site Scripting (XSS) and Injection attacks, mastering both offensive and defensive techniques.
- Examine risk factors in XML processing, file and software uploads, and deserialization, along with strategies for risk mitigation.
- Get acquainted with key security tools, from code scanners to web application firewalls, while also exploring server and infrastructure hardening techniques.

If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate.

## Audience

This is an **intermediate level** Java programming course, designed for experienced Java Web developers, software engineers, and architects who are seeking to enhance their knowledge and skills in application security, bug hunting, and secure software development. The course would also be well-suited for IT professionals, such as security analysts, security engineers, and DevOps team members, who are responsible for ensuring the security and integrity of web applications in their organizations.

## Pre-Requisites

{{code}}                    {{title}}

## Agenda

### Bug Hunting Foundation

#### Why Hunt Bugs?

- The Language of Cybersecurity
- The Changing Cybersecurity Landscape
- AppSec Dissection of SolarWinds
- The Human Perimeter
- First Axiom in Web Application Security Analysis
- First Axiom in Addressing ALL Security Concerns

#### Safe and Appropriate Bug Hunting/Hacking

- Warning to All Bug Hunters
- Working Ethically
- Respecting Privacy
- Bug/Defect Notification
- Bug Hunting Pitfalls

### Moving Forward From Hunting Bugs

#### Removing Bugs

- Open Web Application Security Project (OWASP)
- OWASP Top Ten Overview
- Web Application Security Consortium (WASC)
- Common Weaknesses Enumeration (CWE)
- CERT Secure Coding Standard
- Microsoft Security Response Center
- Software-Specific Threat Intelligence

### Bug Stomping 101

#### Unvalidated Data

- CWE-787, 125, 20, 416, 434, 190, 476 and 119
- Potential Consequences
- Defining and Defending Trust Boundaries
- Rigorous, Positive Specifications
- Allow Listing vs Deny Listing
- Challenges: Free-Form Text, Email Addresses, and Uploaded Files

### **A01: Broken Access Control**

- CWE-22, 352, 862, 276, and 732
- Elevation of Privileges
- Insufficient Flow Control
- Unprotected URL/Resource Access/Forceful Browsing
- Metadata Manipulation (Session Cookies and JWTs)
- Understanding and Defending Against CSRF
- CORS Misconfiguration Issues

### **A02: Cryptographic Failures**

- CWE-200
- Identifying Protection Needs
- Evolving Privacy Considerations
- Options for Protecting Data
- Transport/Message Level Security
- Weak Cryptographic Processing
- Keys and Key Management
- NIST Recommendations

### **A03: Injection**

- CWE-79, 78, 89, and 77
- Pattern for All Injection Flaws
- Misconceptions With SQL Injection Defenses
- Drill Down on Stored Procedures
- Other Forms of Server-Side Injection
- Minimizing Server-Side Injection Flaws
- Client-side Injection: XSS
- Persistent, Reflective, and DOM-Based XSS
- Best Practices for Untrusted Data

### **A04: Insecure Design**

- Secure Software Development Processes
- Shifting Left
- Principles for Securing All Designs
- Leveraging Common AppSec Practices and Control
- Paralysis by Analysis
- Actionable Application Security
- Additional Tools for the Toolbox

### **A05: Security Misconfiguration**

- System Hardening: IA Mitigation
- Risks with Internet-Connected Resources
- Minimalist Configurations
- Application Allow Listing
- Secure Baseline
- Segmentation with Containers and Cloud
- CWE-611
- Safe XML Processing

### **Bug Stomping 102**

#### **A06: Vulnerable and Outdated Components**

- Problems with Vulnerable Components
- Software Inventory
- Managing Updates: Balancing Risk and Timeliness
- Virtual Patching
- Dissection of Ongoing Exploits

#### **A07: Identification and Authentication Failures**

- CWE-306, 287, 798 and 522
- Quality and Protection of Authentication Data
- Anti-Automation Defenses
- Multifactor Authentication
- Proper Hashing of Passwords
- Handling Passwords on Server Side

#### **A08: Software and Data Integrity Failures**

- CWE-502
- Software Integrity Issues and Defenses
- Using Trusted Repositories
- CI/CD Pipeline Issues
- Protecting Software Development Resources
- Serialization/Deserialization

#### **A09: Security Logging and Monitoring Failures**

- Detecting Threats and Active Attacks
- Best Practices for Logging and Logs

- Safe Logging in Support of Forensics

### **A10: Server Side Request Forgeries (SSRF)**

- CWE-918
- Understanding SSRF
- Remote Resource Access Scenarios
- Complexity of Cloud Services
- SSRF Defense in Depth
- Positive Allow Lists

### **Moving Forward with Application Security**

#### **Applications: What Next?**

- Common Vulnerabilities and Exposures
- CWE Top 25 Most Dangerous SW Errors
- Strength Training: Project Teams/Developers
- Strength Training: IT Organizations

### **Secure Development Lifecycle (SDL)**

#### **SDL Overview**

- Attack Phases: Offensive Actions and Defensive Controls
- Secure Software Development Processes
- Shifting Left
- Actionable Items Moving Forward

#### **SDL In Action**

- Risk Escalators
- Risk Escalator Mitigation
- SDL Phases
- Actions for each SDL Phase
- SDL Best Practices

#### **Next Steps**

- Your Secure Coding Action Plan
- Key Resources

### **Follow On Courses**

{{code}}

{{title}}

## Related Courses

TT8700	Securing Databases   Database Security
TT8800	Information Assurance (STIG) Overview
TTAI2810	Mastering Machine Learning Operations (MLOps) and AI Security Boot Camp
TTAI2820	Mastering AI Security Boot Camp
TTAI2832	Applying AI to the 2021 OWASP Top Ten
TTAI2835	AI Secure Programming for Web Applications / Technical Overview
TT8320-J	Java Secure Coding Camp   Attacking and Securing Java Web Applications
TT8120	Securing Web Applications   2021 OWASP Top Ten and Beyond

## Attend a Course

Please feel free to Register Online or call 844-475-4559 toll free to connect with our Registrar for assistance. If you ever need additional date options, please [contact us](#) for scheduling.