Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J)

Category: Software Development & Programming | **Vendor:** Trivera Tech

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

Attacking and Securing Java EE Web Applications is a lab-intensive, hands-on Java EE security training course that provides a unique coverage of Java application security. In this course, students begin with penetration testing, hunting for bugs in Java web applications. They then thoroughly examine best practices for defensively coding web applications, covering all the OWASP Top Ten as well as several additional prominent vulnerabilities (such as file uploads, CSRF and direct object references). Students will repeatedly attack and then defend various assets associated with fully functional web applications and services. This hands-on approach drives home the mechanics of how to secure JEE web applications in the most practical of terms.Students will leave the course armed with the skills required to recognize actual and potential software vulnerabilities and implement defenses for those vulnerabilities. This course begins by developing the skills required to fingerprint a web application and then scan it for vulnerabilities and bugs. Practical labs using current tools and techniques provide students with the experience needed to begin testing their own applications.

About This Course

Attacking and Securing Java EE Web Applications is a lab-intensive, hands-on Java EE security training course that provides a unique coverage of Java application security. In this course, students begin with penetration testing, hunting for bugs in Java web applications. They then thoroughly examine best practices for defensively coding web applications, covering all the OWASP Top Ten as well as several additional prominent vulnerabilities (such as file uploads, CSRF and direct object references). Students will repeatedly attack and then defend various assets associated with fully functional web applications and services. This hands-on approach drives home the mechanics of how to secure JEE web applications in the most practical of terms.Students will leave the course armed with the skills required to recognize actual and potential software vulnerabilities and implement defenses for those vulnerabilities. This course begins by developing the skills required to fingerprint a web application and then scan it for vulnerabilities and bugs. Practical labs using current tools and techniques provide students with the experience needed to begin testing their own applications.

Who Should Attend

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. Familiarity with Java and Java EE is required and real world programming experience is highly recommended. Ideally students should have approximately 6 months to a year of Java and JEE working knowledge.

Prerequisites & Entry Requirements

General Prerequisites:

Practical hands-on Java web development experience. This is java coding class that requires intermediate Java developer skills to complete the lab work.

Learning Outcomes

Upon successful completion of this course, participants will be able to:

Working in a dynamic, lab-intensive hands-on coding environment you'll learn to:
Ensure that any bug hunting is performed in a safe and appropriate manner
Identify defect/bug reporting mechanisms within their organizations
Work with specific tools for targeted vulnerabilities
Avoid common mistakes that are made in bug hunting and vulnerability testing
Understand the concepts and terminology behind defensive, secure coding including the phases and goals of a typical exploit
Develop an appreciation for the need and value of a multilayered defense in depth
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
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
Detect, attack, and implement defenses against XSS and Injection attacks
Understand the risks associated with XML processing, file uploads, and server-side interpreters and how to best eliminate or mitigate those risks
Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure

Detailed Course Outline

Why Hunt Bugs

- Security and Insecurity
- Dangerous Assumptions
- Attack Vectors

Safe and Appropriate Bug Hunting/Hacking

- Working Ethically
- Respecting Privacy
- Bug/Defect Notification
- Bug Bounty Programs

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 Resources

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 Eclipse (JEE Version) and Apache TomEE 7x
- Tutorial: Working with the HSQL Database

Unvalidated Data

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

A1: Injection

- Injection Flaws
- SQL Injection Attacks Evolve
- Drill Down on Stored Procedures
- Other Forms of Injection
- Minimizing Injection Flaws

A2: Broken Authentication

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

A3: Sensitive Data Exposure

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

A4: XML External Entities (XXE)

- XML Parser Coercion
- XML Attacks: Structure
- XML Attacks: Injection
- Safe XML Processing

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

A6: Security Misconfiguration

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

A7: Cross Site Scripting (XSS)

- XSS Patterns
- Persistent XSS
- Reflective XSS
- DOM-Based XSS
- Best Practices for Untrusted Data

A8/9: Deserialization/Vulnerable Components

- Deserialization Issues
- Identifying Serialization and Deserializations
- Vulnerable Components
- Software Inventory
- Managing Updates

A10: Insufficient Logging and Monitoring

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

Spoofing, CSRF, and Redirects

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

SDL Overview

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

Applications: 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 Control

Making Application Security Real

- Cost of Continually Reinventing
- Paralysis by Analysis
- Actional Application Security
- Additional Tools for the Toolbox

, 1. 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

2. 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

3. 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

4. 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

5. 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

6. 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

7. 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

8. 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

9. 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

10. A06: Vulnerable and Outdated Components

Problems with Vulnerable Components

Software Inventory

Managing Updates: Balancing Risk and Timeliness

Virtual Patching

Dissection of Ongoing Exploits

11. 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

12. 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

13. A09: Security Logging and Monitoring Failures

Detecting Threats and Active Attacks

Best Practices for Logging and Logs

Safe Logging in Support of Forensics

14. 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

15. Applications: What Next

Common Vulnerabilities and Exposures

CWE Top 25 Most Dangerous SW Errors

Strength Training: Project Teams/Developers

Strength Training: IT Organizations

16. Secure Development Lifecycle (SDL)

17. SDL Overview

Attack Phases: Offensive Actions and Defensive Controls

Secure Software Development Processes

Shifting Left

Actionable Items Moving Forward

18. SDL In Action

Risk Escalators

Risk Escalator Mitigation

SDL Phases

Actions for each SDL Phase

SDL Best Practices

Additional Course Details

Nexus Humans Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J) training program is a workshop that presents an invigorating mix of sessions, lessons, and masterclasses meticulously crafted to propel your learning expedition forward.

This immersive bootcamp-style experience boasts interactive lectures, hands-on labs, and collaborative hackathons, all strategically designed to fortify fundamental concepts.

Guided by seasoned coaches, each session offers priceless insights and practical skills crucial for honing your expertise. Whether you're stepping into the realm of professional skills or a seasoned professional, this comprehensive course ensures you're equipped with the knowledge and prowess necessary for success.

While we feel this is the best course for the Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J) course and one of our Top 10 we encourage you to read the course outline to make sure it is the right content for you.

Additionally, private sessions, closed classes or dedicated events are available both live online and at our training centres in Dublin and London, as well as at your offices anywhere in the UK, Ireland or across EMEA.

Frequently Asked Questions

Q: What delivery options are available for Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J)?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 4-day Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J) course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J) training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J)?

Yes, we provide corporate training, dedicated training, and closed classes for Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J). Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: Java; JEE ; Application Security

---

Q: Why choose Nexus Human for Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J)?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Java EE Secure Coding Camp | Attacking & Securing Java / Jakarta JEE Web Applications (TT8320-J) training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 XXX XXXX (Ireland) | +44 20 XXXX XXXX (UK)