DEV541: Secure Coding in Java/JEE: Developing Defensible Applications

Take this course to learn how to build secure Java applications and gain the knowledge and skills to:

- Keep your website from getting hacked
- Avoid becoming the next headline
- Counter a wide range of application attacks
- Prevent critical security vulnerabilities that can lead to data loss
- Understand the attackers mindset and how your applications can be hacked

This course teaches you the art of modern web defense for Java applications by focusing on foundational defensive techniques, cutting edge protections, and Java EE security features that you can use in your applications as soon as you return to work. This includes learning how to:

- Identify security defects in your code
- Fix security bugs using secure coding techniques
- Utilize secure HTTP headers to prevent attacks
- Secure your sensitive REST services
- Incorporate security into your development process
- Use freely available security tools to test your applications

Great developers have traditionally distinguished themselves by the elegance, effectiveness, and reliability of their code. That's still true, but elegance, effectiveness, and reliability have now been joined by security. This unique SANS course allows you to bone up on the skills and knowledge required to prevent your applications from getting hacked.

How the course works?

This is a comprehensive course covering a huge set of skills and knowledge. It's not a high-level theory course. It's about real, hands-on programming. In this course you will examine actual code, work with real tools, build applications, and gain confidence in the resources you need for the journey to improving the security of Java applications.

Rather than teaching students to use a set of tools, we're teaching students concepts of secure programming. This involves looking at a specific piece of code, identifying a security flaw, and implementing a fix for flaws found on the OWASP Top 10 and CWE/SANS Top 25 Most Dangerous Programming Errors.

The class culminates in a Secure Development Challenge where you perform a security review of a real-world open source application. You will conduct a code review, perform security testing to actually exploit real vulnerabilities, and finally, using the secure coding techniques that you have learned in class, implement fixes for these issues.

PCI Compliance

Section 6.5 of the Payment Card Industry (PCI) Data Security Standard (DSS) instructs auditors to verify that processes exist that require training in secure coding techniques for developers. If your Java application processes cardholder data and you are required to meet PCI compliance then this course is for you.

WRITE ONCE SECURELY, RUN ANYWHERE

Who Should Attend

- Developers who want to build more secure applications
- Java EE programmers
- Software engineers
- Software architects
- Developers who need to be trained in secure coding techniques to meet PCI compliance

This class is focused specifically on software development but is accessible enough for anyone who's comfortable working with code and has an interest in understanding the developer's perspective including:

- Application security auditors
- Technical project managers
- Senior software QA specialists
- Penetration testers who want a deeper understanding of target applications or who want to provide more detailed vulnerability remediation options

You Will Be Able To

- Use a web application proxy to view and manipulate HTTP requests and responses.
- Review and perform basic exploits of common web application vulnerabilities, such as those found in the SANS/CWE Top 25 and the OWASP Top 10:
  - Cross-Site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
  - SQL Injection
  - Parameter Manipulation
  - Open Redirect
  - Session Hijacking
  - Clickjacking
  - Authentication & Access Control Bypass
- Mitigate common web application vulnerabilities using secure coding practices and Java libraries, including the following:
  - Input Validation
  - Blacklist & Whitelist Validation
  - Regular Expressions
  - Output Encoding
  - Content Security Policy (CSP)
  - Client-Side Security Headers
- Build applications using the following:
  - Java EE Authentication
  - Basic and Forms based authentication
  - Client certificates
  - SSL/TLS
  - Java Secure Sockets Extension (JSSE)
  - Secure password storage techniques
  - Java Cryptography Architecture (JCA)
  - SecurityManager
- Implement a secure software development lifecycle (SDLC) including code review, static analysis, and dynamic analysis techniques

541.1 HANDS ON: Data Validation

Improper data validation is the root cause of the most prevalent web application vulnerabilities today. You will learn about some of the most prevalent web applications vulnerabilities such as XSS, CSRF, SQL Injection, HTTP Response Splitting, and Parameter Manipulation. You will see how to find these issues and how to recreate them in a running application. Then you will use a variety of methods to actually fix these vulnerabilities in your Java code. The course is full of hands on exercises where you can apply practical data validation techniques that you can use to prevent common attacks with defense ranging from input validation, output encoding, and use of new techniques like Content Security Policy.

Topics: Web Application Attacks; Cross-Site Scripting (XSS); Cross-Site Request Forgery (CSRF); SQL Injection; HTTP Response Splitting; Parameter Manipulation; Directory Traversal; Web Application Proxies; Validation Concerns; Character Encoding; Input Validation; Output Encoding; Blacklisting & Whitelisting; Validation Techniques; Regular Expressions; Servlet Filters; Output Encoding; Content Security Policy; Prepared Statements; CSRF Defense

541.2 HANDS ON: Authentication and Session Management

Broken authentication and session management are common issues that can compromise the integrity of your system. Weak authentication protections can allow an attacker to expose your most sensitive secrets: your data! You will learn about these vulnerabilities and what you can do to design and code stronger authentication protections from the start. You will learn how to use Java EE Container Based Authentication and setup Basic, Form Based and client certificate authentication. You will also learn how to protect data in transit using SSL and how to securely store passwords at rest. Various authorization attacks will be discussed as well as unvalidated forwards and redirects. Session management attacks and defenses are covered in addition to Clickjacking and associated defenses.

Topics: Authentication Factors; Authentication Attacks; Java EE Authentication; Basic Authentication; Form Based Authentication; Client certificates; Using SSL; Secure password storage; Authorization; Web and EJB access control; Authorization Attacks; Access control bypass; Unvalidated forwards and redirects; State Management Attacks; Session hijacking; Session fixation; Clickjacking; Using X-Frame-Options

541.3 HANDS ON: Java Platform and API Security

Java is the language of choice for the development of many mission critical applications. As such, it is vital to understand the security features and implications of using the Java language itself and the Java Runtime Environment (JRE). Through numerous hands-on exercises you will learn about the Security Manager, how code privileges are managed, and how to sign jar files. You will also learn about Exception handling and try/catch/finally blocks as well as the importance of logging. With hands-on exercises you will also write code to encrypt both data in transit and data at rest using the Java Secure Socket Extension (JSSE) and the Java Cryptography Architecture (JCA) as well as String immutability, integer and double overflows, and about numerous Java language features that you should consider while writing secure code.

Topics: Java Security Manager; Permissions; Policy file; Jar signing; Class security; Error Handling; Exceptions; Using try/catch/finally; Logging; Logging frameworks; ESAPI logging; Encryption; Java Secure Sockets Extension (JSSE); Java Cryptography Architecture (JCA); Integer and Double Overflows; Thread safety; Race Conditions; Web Service (JAX-RS) Security; REST Security; OAuth

541.4 HANDS ON: Secure Development Lifecycle

Using what you have learned about web application vulnerabilities, you will conduct a security review of a real-world open source application. You will see first hand how to integrate security in your software development life cycle (SDLC) by first conducting a code review of a large, widely used open source application. Once you have identified various vulnerabilities in the code itself you will then perform security testing and actually exploit these weaknesses. Once they have been exploited you will then fix them using the secure coding techniques you have learned in class. The Secure Development Challenge introduces you to what is needed in a Secure SDLC and shows you how to do it first hand!

Topics: Security and the SDLC; Conducting a secure code review; Manual code review; Using a static analysis tool; Using FindBugs; Integrating code review into the SDLC; Security Testing; Exploiting XSS, CSRF, and SQL Injection; Secure Coding; Fixing weaknesses in a running application

DEV541 Training Formats

(subject to change)

Live Training
sans.org/security-training/by-location/all

Community SANS Events
sans.org/community

Community SANS
sans.org/community

OnSite
sans.org/onsite

OnDemand
sans.org/ondemand

SelStudy
sans.org/selfstudy

"I never thought I could learn so much in such short time without feeling burned out. Great job making it engaging and interesting."

-Jeff Eubanks, Mainstream Engineering Corp.