Secure Java Development

Course Summary

Description

This advanced course shows experienced developers of Java enterprise applications how to secure those applications and to apply best practices with regard to secure coding. Authentication, authorization, and input validation are major themes, and participants use cryptographic algorithms (via the JCA and JCE) for specific development scenarios. The course also includes thorough discussions and hands-on exercises in common web-application concerns and hacks (XSS, injection, etc.), HTTPS configuration and certificate management, error handling, logging, and auditing. Security for XML-based web services is also covered, including discussions of XML cryptography, WS-Security, and single-sign-on scenarios under SAML 2.0.

Objectives
At the end of this course, students will be able to:

- Be prepared to develop secure Java web applications, or to secure existing applications by refactoring as necessary.
- Instruct the web container to enforce your desired authentication and authorization policies.
- Validate user input aggressively, for general application health and to foil specific attacks including injection and XSS.
- Be on guard against common web-application attack vectors such as cross-site scripting, injection, and request forgery, and take appropriate counter-measures.
- Configure a server and/or application to use one-way or two-way HTTPS.
- Use cryptography appropriately in application design and implementation, including signature and signature verification, encryptiong, hashing, and tactical use of securerandom numbers.
- Secure log files and establish audit trails for especially sensitive information or actions.
- Be aware of appropriate techniques and standards for web-services security, including XML cryptography, WS-Security, and SAML.

Topics

- Java SE Security
- Code Signature and Key Management
- Secure Development Practices
- Java EE Security
- Authentication and Authorization
- Special Concerns for Web Applications
- HTTPS and Certificates
- The Java Secure Sockets Extension
- Cryptography in Web Applications
- Secure Development Practices
- Securing the Service-Oriented Enterprise
- XML Signature
- XML Encryption
- SAML Bindings
- Federated Identity

Audience

This course is appropriate for experienced Java developers who want to be able to follow secure development practice and to secure new and existing standalone, web, and enterprise applications.

Prerequisites

Java programming experience is essential, including understanding of OO practices, exception handling and multi-threading. Servlets programming experience is recommended but not required. JSP page-authoring experience is recommended but not required. Experience developing JAX-RPC or JAX-WS web services is strongly recommended for the final chapters -- but not strictly required.

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Secure Java Development

Course Outline

I. Java SE Security
- A. Holistic Security Practices
- B. Threats to the User
- C. The Class Loader and Bytecode Verifier
- D. System Classes and the Core API
- E. SecurityManager and AccessController
- F. Permissions
- G. Implication
- H. CodeSources
- I. Policies
- J. Configuring Java SE Security
- K. Dynamic Policies
- L. Privileged Actions

II. Code Signature and Key Management
- A. Encryption and Digital Signature
- B. Keystores
- C. Keys and Certificates
- D. Certificate Authorities
- E. The KeyStore API
- F. Signing JARs
- G. Signed CodeSources
- H. Additional Policy Semantics

III. Secure Development Practices
- A. Code Injection
- B. Final Classes and Methods
- C. Singletons, Factories, and Flyweights
- D. Methods, Collections, and Data Hiding
- E. Sealing JARs
- F. Code Obfuscation
- G. Object Serialization

IV. Java EE Security
- A. Threats and Attack Vectors
- B. Server, Network, and Browser Vulnerabilities
- C. Secure Design Principles
- D. GET vs. POST
- E. Container Authentication and Authorization
- F. HTML Forms
- G. Privacy Under /WEB-INF
- H. HTTP and HTTPS
- I. Other Cryptographic Practices
- J. SOA and Web Services
- K. The OWASP Top 10

V. Authentication and Authorization
- A. HTTP BASIC and DIGEST Authentication Schemes
- B. Declaring Security Constraints
- C. User Accounts
- D. Safeguarding Credentials in Transit
- E. Replay Attacks
- F. Authorization Over URL Patterns
- G. Roles
- H. FORM Authentication
- I. Login Form Design
- J. EJB Authorization
- K. Programmatic Security
- L. Programmatic Security in JSF

VI. Special Concerns for Web Applications
- A. Single Points of Decision
- B. Cross-Site Scripting
- C. Validation vs. Output Escaping
- D. Forceful Browsing
- E. Cross-Site Request Forgery
- F. Request Tokens
- G. Injection Attacks
- H. Protections in JDBC and JPA
- I. Session Management
- J. Taking Care of Cookies
- K. Validating User Input
- L. Validation Practices
- M. Whitelist Validation
- N. Blacklist Validation
- O. Regular Expressions
- P. Constructing REs
- Q. Using REs in Java
- R. JSF Validation

VII. HTTPS and Certificates
- A. Digital Cryptography
- B. Encryption
- C. SSL and Secure Key Exchange
- D. Hashing
- E. Signature
- F. Keystores
- G. keytool
- H. Why Keys Aren't Enough
- I. X.509 Certificates
- J. Certificate Authorities
- K. Obtaining a Signed Certificate
- L. Configuring HTTPS
- M. Client-Side Certificates and Two-Way SSL
- N. PKCS #12 and Trust Stores
- O. CLIENT-CERT Authentication

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Secure Java Development

Course Outline (cont'd)

VIII. The Java Secure Sockets Extension
- A. The Role of JSSE
- B. Support in the URL Class
- C. Support in Socket and ClientSocket Classes
- D. Configuring JSSE
- E. Customizing JSSE

IX. Cryptography in Web Applications
- A. Secure Random Number Generation
- B. The MessageDigest Class
- C. The Java Cryptography Extensions
- D. The SecretKey and KeyGenerator Types
- E. The Cipher Class
- F. Choosing Algorithms and Key Sizes
- G. Dangerous Practices
- H. Using JSSE

X. Secure Development Practices
- A. Secure Development Cycle
- B. Error Handling and Information Leakage
- C. Failing to a Secure Mode
- D. Logging Practices
- E. Appropriate Content for Logs
- F. Auditing
- G. Strategies: Filters, Interceptors, and Command Chains
- H. Penetration Testing
- I. Back Doors

XI. Securing the Service-Oriented Enterprise
- A. Security for Web Services
- B. Threats
- C. CIA Goals
- D. Solution Levels: W3C, OASIS, Java EE
- E. Scenario: Secure Multi-Party Conversation
- F. Cryptography
- G. WS-Security and WS-SecurityPolicy
- H. Scenario: Sharing Security Information
- I. SAML and XACML
- J. Scenario: Multiple User Realms
- K. Scenario: Single Sign-On
- L. Technology Stacks: WS-Federation and Liberty Alliance
- M. The WS-I Basic Security Profile

XII. XML Signature
- A. Use Case: Non-Repudiation
- B. XML Digital Signature
- C. Cryptography Backgrounder
- D. Canonical XML
- E. Enveloped, Enveloping, and Detached Signatures
- F. SignedInfo and References
- G. The Java Cryptography Architecture
- H. Keystores
- I. Why Keys Aren't Enough
- J. X.509 Certificates and Certificate Chains
- K. The KeyStore API
- L. Java XML Digital Signature API
- M. Steps to Sign and Verify XML Content
- N. JAX-WS Message Handlers
- O. Foiling the Man in the Middle

XIII. XML Encryption
- A. Use Case: Confidentiality
- B. XML Encryption
- C. EncryptedData
- D. Element vs. Content Encryption
- E. Key Wrapping
- F. The Java Cryptography Extensions
- G. Apache XML Security
- H. Steps to Encrypt and Decrypt XML Content
- I. Choosing Algorithms and Key Sizes

XIV. SAML Bindings
- A. Use Case: Speaking "Through" the Browser
- B. The SOAP Binding
- C. SAML Over HTTP
- D. The Browser as Messenger
- E. The Redirect, POST, and Artifact Bindings
- F. The PAOS Binding
- G. The URI Binding

XV. Federated Identity
- A. What is Federation?
- B. Problems for Identity Federation
- C. SAML 2.0 Federations
- D. Single Sign-On
- E. Account Linking and Persistent Pseudonyms
- F. Transient Pseudonyms
- G. Name ID Mapping
- H. Federation Termination
- I. OpenSSO
- J. Fedlets

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically