

Secure Coding Attack Lab

Application-security testing, remediation and retesting in a controlled lab

FA-0686 · 2 days

Cybersecurity · Software Development

Introduction

This two-day workshop connects application-security testing with practical secure coding. Participants map a supplied application, reproduce selected vulnerabilities, inspect their causes, apply corrections and collect retest evidence.

Injection, unsafe data handling, business logic, cryptography, browser security and access control are examined through selected exercises and demonstrations. Coverage relates to secure-development concerns in PCI DSS Requirement 6; this independent workshop is not a PCI compliance assessment or certification.

Learning outcomes

- Prepare a controlled lab and map application and API trust boundaries.
- Reproduce selected injection, data-handling and business-logic weaknesses in the supplied application.
- Explain password, secret, cryptographic, XSS and CSRF risks and appropriate defensive controls.
- Test authentication, session and object-level authorisation using supplied test accounts.
- Apply selected code corrections, perform negative and regression tests, and document evidence.

Prerequisites

- Basic software-development or application-support experience
- Familiarity with HTTP requests, responses and web applications
- Basic understanding of APIs and database-backed applications
- Ability to read code in at least one common programming language

- Basic command-line skills
- Administrator access to the supplied lab workstation or virtual machine
- A laptop capable of running the designated lab environment
- No previous penetration-testing experience required

Training outline

1. Day 1 — Lab setup and attack-surface mapping

1.1 Lab Environment

- Vulnerable application overview
- Local application and database services
- Test users and roles
- Application reset and recovery

1.2 Security Testing Tools

- Browser developer tools
- Intercepting proxy
- API client
- Command-line request tools
- Source-code editor

1.3 Attack-Surface Mapping

- Web routes
- API endpoints
- HTTP methods
- Parameters
- Headers
- Cookies
- Request bodies
- Authentication boundaries
- Role and permission boundaries
- Client-side trust boundaries

2. Day 1 – Injection and defensive query handling

2.1 SQL Injection

- Query parameter manipulation
- Form-input manipulation
- Authentication query weaknesses
- Unsafe dynamic queries
- Parameterised query remediation
- Remediation verification

2.2 Command Injection

- Unsafe operating-system command construction
- Shell metacharacter handling
- Argument injection
- Safe process execution
- Command allow-listing
- Corrective testing

2.3 LDAP and XPath Injection

- Directory-search filter manipulation
- XPath query manipulation
- Unsafe query construction
- Input escaping
- Safe query handling
- Guided demonstration

2.4 Secure Coding Controls

- Input validation
- Parameterised operations
- Allow-listing
- Least-privilege database access
- Negative testing
- Regression testing

3. Day 1 — Data, object binding and file handling

3.1 Unsafe Input and Data Handling

- Missing validation
- Malformed input
- Boundary-value manipulation
- Invalid data types
- Truncation risks
- Secure error handling

3.2 Mass Assignment and Object Binding

- Automatic request-to-object mapping
- Hidden properties
- Privileged properties
- Over-posting
- Data-transfer objects
- Explicit property mapping
- Property-level authorisation

3.3 Insecure Deserialization

- Untrusted serialized input
- Unsafe object reconstruction
- Polymorphic object handling
- Trusted-type allow-listing
- Schema-restricted formats
- Secure replacement patterns

3.4 File and Data Processing

- File-name manipulation
- Path traversal
- Content-type trust
- File-size validation
- Temporary-file handling

- Secure storage boundaries

3.5 Buffer, Pointer and Shared-Data Risks

- Buffer-boundary weaknesses in relevant native-code examples
- Integer conversion risks
- Null-reference and error-handling failures
- Shared mutable state
- Race-condition concepts
- Defensive validation

4. Day 1 — Business processes and API validation

4.1 Business-Process Mapping

- Intended transaction sequence
- Required approvals
- State transitions
- Trust assumptions
- Server-side enforcement points

4.2 Workflow Manipulation

- Step skipping
- Out-of-order operations
- Request replay
- Duplicate submissions
- Partial-completion abuse

4.3 Parameter and Value Manipulation

- Price manipulation
- Quantity manipulation
- Discount manipulation
- Negative values
- Extreme values
- Hidden-field manipulation

- Server-side value calculation

4.4 Client-Side Validation Bypass

- Disabled-control bypass
- Front-end role-check bypass
- Client-calculated values
- Exposed configuration
- Server-side control enforcement

4.5 API Manipulation

- Request parameter tampering
- HTTP method manipulation
- Request-body modification
- Hidden endpoint access
- Server-side validation

5. Day 2 – Cryptography, passwords and secrets

5.1 Weak Cryptographic Implementations

- Deprecated algorithms
- Weak cipher modes
- Inadequate key lengths
- Improper random-number generation
- Incorrect IV generation or reuse for the selected cryptographic mode
- Reused nonces

5.2 Password Protection

- Plaintext password storage
- Reversible password storage
- Weak password hashes
- Password-hashing functions
- Salt generation
- Work-factor configuration

5.3 Hard-Coded Credentials and Secrets

- Embedded passwords
- API keys in source code
- Configuration-file secrets
- Log and diagnostic leakage
- Secret scanning
- Secure secret injection

5.4 Encryption in Transit

- TLS configuration
- Certificate validation
- Hostname verification
- Trust-store handling
- Insecure protocol fallback

5.5 Token and Randomness Weaknesses

- Predictable tokens
- Weak session identifiers
- Improper signing
- Missing integrity checks
- Secure token generation

5.6 Cryptographic Remediation

- Algorithms and parameters appropriate to the use case and applicable policy
- Secure cryptographic libraries
- Key separation
- Secret rotation
- Secure failure handling
- Remediation verification

6. Day 2 — Browser security and business logic

6.1 Communication Protocols and Channels

- HTTP method restrictions
- Header trust
- Host-header manipulation
- Content-type enforcement
- Redirect handling
- Secure service communication

6.2 Client-Side Functionality Abuse

- JavaScript function invocation
- Local storage exposure
- Browser token storage
- Source-map exposure
- Client-side secret exposure
- Server-side validation requirements

6.3 Cross-Site Scripting

- Reflected XSS
- Stored XSS
- DOM-based XSS
- HTML-context injection
- Attribute-context injection
- Unsafe DOM operations
- Context-aware output encoding
- HTML sanitisation
- Content Security Policy as defence in depth, not a substitute for encoding or sanitisation

6.4 Cross-Site Request Forgery

- State-changing request discovery
- Cookie-based authentication
- Forged form submissions

- Anti-CSRF tokens
- SameSite cookies as an additional CSRF defence, not a universal replacement for request protection
- Origin validation
- Referer validation
- Re-authentication for sensitive actions

6.5 Server-Side Enforcement

- Trust-boundary validation
- Request integrity checks
- Server-side business rules
- Secure transaction validation
- Regression testing

7. Day 2 – Authentication, sessions and access control

7.1 Identification, Authentication and Authorisation

- User identification
- Credential validation
- Role verification
- Permission enforcement
- Policy decision points
- Policy enforcement points

7.2 Authentication Weaknesses

- Default credentials
- Username enumeration
- Password-reset weaknesses
- Multi-factor bypass
- Account-lockout weaknesses
- Missing rate limiting

7.3 Session and Token Weaknesses

- Session fixation

- Session-token predictability
- Token reuse
- Missing logout invalidation
- Missing session rotation
- Weak cookie attributes
- Improper token validation

7.4 Horizontal Privilege Escalation

- User-to-user data access
- Identifier substitution
- Cross-account actions
- Resource ownership checks
- Tenant-isolation testing

7.5 Vertical Privilege Escalation

- User-to-administrator access
- Restricted function invocation
- Role-parameter manipulation
- Privileged route access
- Server-side permission enforcement

7.6 IDOR and BOLA

- Sequential identifiers and the need for authorisation regardless of identifier format
- Predictable or opaque identifiers: neither replaces per-object authorisation
- Object-identifier manipulation
- Read access testing
- Update access testing
- Delete access testing
- Per-object authorisation

7.7 Access-Control Remediation

- Centralised authorisation
- Deny-by-default policies

- Consistent middleware enforcement
- Role-permission matrices
- Negative authorisation testing
- Privilege regression testing

8. Day 2 — Remediation, retesting and evidence

8.1 Vulnerability Triage

- Reproduction reliability
- Attack preconditions
- Exploitability
- Data exposure
- Business impact
- Root-cause classification

8.2 Secure Coding Corrections

- Input validation
- Parameterised operations
- Context-aware output encoding
- Explicit object binding
- Centralised authorisation
- Secure cryptographic APIs
- Safe error handling
- Least-privilege execution

8.3 Fix Validation

- Original attack replay
- Negative testing
- Boundary testing
- Alternate payload testing
- Role-based regression testing
- API regression testing

8.4 Remediation Evidence

- Vulnerability description
- Reproduction steps
- Root cause
- Corrective action
- Retest results
- Evidence capture