Web application security

CYDWeb3d | 3 days | On-site or online | Hands-on

Your application written in any programming language works as intended, so you are done, right? But did you consider feeding in incorrect values? 16Gbs of data? A null? An apostrophe? Negative numbers, or specifically -1 or -2^31? Because that's what the bad guys will do – and the list is far from complete.

Handling security needs a healthy level of paranoia, and this is what this course provides: a strong emotional engagement by lots of hands-on labs and stories from real life, all to substantially improve code hygiene. Mistakes, consequences, and best practices are our blood, sweat and tears.

The curriculum goes through the common Web application security issues following the OWASP Top Ten but goes far beyond it both in coverage and the details.

All this is put in the context of Java, and extended by core programming issues, discussing security pitfalls of the Java language.

So that you are prepared for the forces of the dark side.

So that nothing unexpected happens.

Nothing.

Skills and drills

24 LABS

17 CASE STUDIES

Audience

Web developers

Group size

12 participants

Preparedness

General Web development

Outline

- Cyber security basics
- The OWASP Top Ten 2021
- Wrap up

Standards and references

OWASP, CWE and Fortify Taxonomy

What you'll have learned

- Getting familiar with essential cyber security concepts
- Understanding how cryptography supports security
- Understanding Web application security issues
- Detailed analysis of the OWASP Top Ten elements
- Putting Web application security in the context of any programming language
- Going beyond the low hanging fruits
- Managing vulnerabilities in third party components

Table of contents

- Random number generation
- Pseudo random number generators (PRNGs)
- Cryptographically secure PRNGs

Lab – Using random numbers

Case study – Equifax credit account freeze

- Confidentiality protection
  - Symmetric encryption
  - Block ciphers
  - Modes of operation
  - Modes of operation and IV – best practices

Lab – Symmetric encryption

  - Asymmetric encryption
  - The RSA algorithm
  - Combining symmetric and asymmetric algorithms
  - Key exchange and agreement
  - Key exchange
  - Diffie-Hellman key agreement algorithm
  - Key exchange pitfalls and best practices

Day 2

› The OWASP Top Ten 2021

A03 - Injection

- Injection principles
- Injection attacks
- SQL injection
  - SQL injection basics

Lab – SQL injection

  - Attack techniques
  - Content-based blind SQL injection
  - Time-based blind SQL injection
  - SQL injection best practices
    - Input validation
    - Parameterized queries

Lab – Using prepared statements

Case study – Hacking Fortnite accounts

- Code injection
  - OS command injection
    - OS command injection best practices

Case study – Shellshock

Lab - Shellshock

- *Case study – Command injection in AVTECH IP cameras*

- HTML injection - Cross-site scripting (XSS)
  - Cross-site scripting basics
  - Cross-site scripting types
    - Persistent cross-site scripting
    - Reflected cross-site scripting
    - Client-side (DOM-based) cross-site scripting
  - *Lab – Stored XSS*
  - *Lab – Reflected XSS*
  - *Case study – XSS in Fortnite accounts*
  - XSS protection best practices
    - Protection principles - escaping
    - *Lab – XSS fix / stored*
    - *Lab – XSS fix / reflected*
    - Additional protection layers – defense in depth

A04 - Insecure Design

- The STRIDE model of threats
- Secure design principles of Saltzer and Schroeder
  - Economy of mechanism
  - Fail-safe defaults
  - Complete mediation
  - Open design
  - Separation of privilege
  - Least privilege
  - Least common mechanism
  - Psychological acceptability
- Client-side security
  - Same Origin Policy
    - Simple request
    - Preflight request
    - Cross-Origin Resource Sharing (CORS)
  - Frame sandboxing
    - Cross-Frame Scripting (XFS) attacks
    - *Lab - Clickjacking*
    - Clickjacking beyond hijacking a click
    - Clickjacking protection best practices
    - *Lab – Using CSP to prevent clickjacking*

A05 - Security Misconfiguration

- Configuration principles
- Server misconfiguration
- Cookie security

- Cookie attributes
- XML entities
  - DTD and the entities
  - Entity expansion
  - External Entity Attack (XXE)
    - File inclusion with external entities
    - Server-Side Request Forgery with external entities
    - *Lab – External entity attack*
    - *Lab – Prohibiting DTD*
    - *Case study – XXE vulnerability in Ivanti products*

Day 3

› The OWASP Top Ten 2021

A06 - Vulnerable and Outdated Components

- Using vulnerable components
- Assessing the environment
- Hardening
- Untrusted functionality import

Case study – The Polyfill.io supply chain attack

- Vulnerability management
  - Patch management
  - Vulnerability management
  - Vulnerability databases
  - DevOps, the CI / CD build process and Software Composition Analysis

A07 - Identification and Authentication Failures

- Authentication
  - Authentication basics
  - Multi-factor authentication (MFA)

Case study – The InfinityGauntlet attack

  - Time-based One Time Passwords (TOTP)
- Session management
  - Session management essentials
  - Why do we protect session IDs – Session hijacking
  - Session fixation
  - Session ID best practices
- Password management
  - Inbound password management

- Storing account passwords
- Password in transit
- *Lab – Is just hashing passwords enough?*
- Dictionary attacks and brute forcing
- Salting
- Adaptive hash functions for password storage
- Password policy
- NIST authenticator requirements for memorized secrets
- Password database migration

- Outbound password management
  - Hard coded passwords
  - Best practices
  - *Lab – Hardcoded password*
  - Protecting sensitive information in memory
  - Challenges in protecting memory
  - *Case study – Microsoft secret key theft via dump files*

A08 - Software and Data Integrity Failures

- Integrity protection
  - Message Authentication Code (MAC)
    - *Lab – Calculating MAC*
  - Digital signature
    - Digital signature with RSA
    - Elliptic Curve Cryptography
    - ECC basics
    - Digital signature with ECC
    - *Lab – Digital signature with ECDSA*
- Subresource integrity
  - Importing JavaScript
  - *Lab – Importing JavaScript*
  - *Case study – The British Airways data breach*
- Insecure deserialization
  - Serialization and deserialization challenges
  - Integrity – deserializing untrusted streams
  - Integrity – deserialization best practices
  - Property Oriented Programming (POP)
    - Creating a POP payload
    - *Lab – Creating a POP payload*
    - *Lab – Using the POP payload*
    - Summary – POP best practices

A09 - Security Logging and Monitoring Failures

- Logging and monitoring principles
- Insufficient logging

Case study – Plaintext passwords at Facebook

- Logging best practices
- Monitoring best practices

A10 - Server-side Request Forgery (SSRF)

- Server-side Request Forgery (SSRF)

Case study – SSRF and the Capital One breach

› Wrap up

Secure coding principles

- Principles of robust programming by Matt Bishop

And now what?

- Software security sources and further reading