JVS101 - DEFENDING JAVASCRIPT

Course Learning Objectives

This course is designed for front-end JavaScript developers. By the end of this course, you'll be able to describe and defend against client-side and server-side cross-site scripting and injection attacks, explain security guidelines for protecting authentication data and implementing access control, protect user sessions, describe and defend against cross-site request forgery and cross-origin sharing, ensure secure communication, and finally, describe best practices for writing secure JavaScript code, protecting data, and implementing a Content Security Policy.

Description

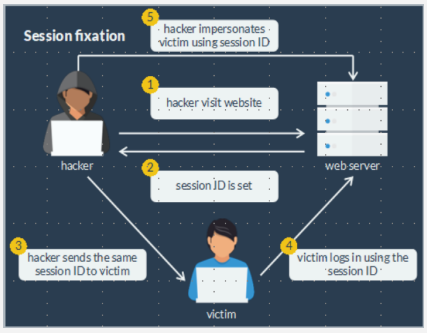
Defending JavaScript is a course for basic and intermediate developers who have some knowledge of application security fundamentals. This course takes a code agnostic approach to secure coding to identify and defend against common risks for front-end JavaScript vulnerabilities. While the focus is on the front-end, there are considerations for back-end security where it applies to the front-end as well. These topics include, cross-site scripting, injection attacks, broken authentication and broken access control, security misconfiguration, and general best practices.

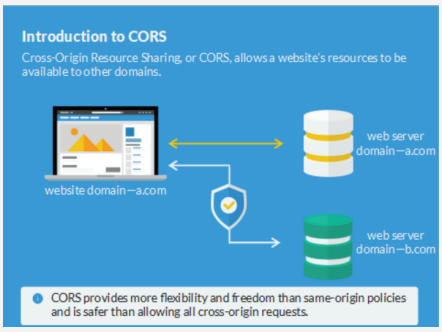
Audience

Basic JavaScript Developers
Intermediate JavaScript Developers

Time Required

Tailored learning - 60 minutes total (approx.)





JVS101 - DEFENDING JAVASCRIPT

Course Outline

1. Introduction

- Why JavaScript
- JavaScript security
- JavaScript vulnerabilities

2. Cross-site scripting

- Introduction to XSS
- Server XSS
- Reflected XSS
- Stored XSS
- Client XSS
- Examples
- About defenses
- Escape data
- Validate input
- Sanitize input
- Review code

3. Injection attacks

- Injection attacks
- Client-side SQL injection
- Example
- NoSQL injection
- Example
- JSON injection
- Example
- XML injection
- Example
- About defenses
- SQL and NoSQL injections
- Best practices for injection attacks
- Best practices for JSON/XML attacks

4. Broken authentication and broken access control

- What is authentication
- Authentication best practices
- Communicating auth. data
- Validating and storing auth. data
- Password policies
- Password reset best practices
- Session management
- Access control

5. Security misconfiguration

- What is CSRF
- Anti-CSRF tokens
- SameSite attribute
- Lax vs Strict mode
- CORS
- Secure communication
- HTTP headers

6. General best practices

- Introduction to data protection
- Remove sensitive information
- Disable where possible
- Dependencies with known vulnerabilities
- Interpreted code integrity
- Content security policy
- Sandboxing

SecurityCompass