RCT201 - DEFENDING REACT

Course Learning Objectives

This course focuses on the React.js framework for JavaScript. Defending React.js takes a look at industry best practices for defending against Cross-site Scripting (XSS), broken authentication, Cross-site Request Forgery (CSRF), and other potential issues such as vulnerable libraries and components.

Description

Defending React.js was created for developers familiar with JavaScript and with limited experience in application security. This course focuses on best practices for addressing the primary threats against applications using the open source library React.js for JavaScript.

Audience

React Developers

Time Required

Tailored learning - 55 minutes total

RCT201 - DEFENDING REACT

Course Outline

1. Introduction

- Introduction
- Why was React created?
- Why use React?
- Benefits of React
- React security
- React vulnerabilities

2. Cross-site scripting

- Introduction
- Reflected XSS
- Stored XSS
- Vulnerable code 1-4
- Client XSS
- Examples
- Vulnerable code 1-3
- About defenses
- Prevention methods
- Vulnerable components & libraries
- Code analysis

3. Broken authentication

- Intro to Authentication
- Intro to Auth0
- Introduction to JWT
- JWT structure
- Security flaws in JWTs
- Unsafe token generation
- Solutions
- Best practices
- Storing JWTs
- Configuration activities
- HttpOnly example
- Authentication best practices

4. Cross-site Request Forgery (CSRF)

- Introduction to CSRF
- Impact of CSRF
- CSRF in the real world
- Code example: Attack sink
- Code example: Attack source
- XSS vs CSRF
- Synchronizer Token Pattern
- Best practices to prevent CSRF
- Example
- Fusion-plugin-csrf-protection-react

5. General best practices

- Introduction to Jscrambler
- Jscrambler integration
- What is Idle Timeout
- Implementation
- Test components: TestUtils
- Test components: Jest
- Example: Snapshot testing
- React test renderer and Jest snapshot
- Output file
- Components with known vulnerabilities
- CSV injection
- Malicious packages
- Improper authorization
- Arbitrary code execution (ACE)
- Prototype Pollution
- On-path attacks
- Defenses

SecurityCompass
COPYRIGHT 2021