API101 - DEFENDING WEB APIS

Course Learning Objectives

Learn about defenses against insufficient Authentication and Authorization in public and private APIs.
Summarize different techniques that defend Web APIs against insufficient input validation and output sanitization.
Discover how to defend against insecure communication and identify best practices for secure communication using SSL and TLS.
Explore ways of reducing the attack surface of Web APIs by implementing defenses against common attacks like CSRF, DoS and DDoS.

Description

This course discusses defenses against the common vulnerabilities of today's RESTful Web APIs. We'll cover the security of connecting to APIs, validating input and output, communication channels, and common attacks.

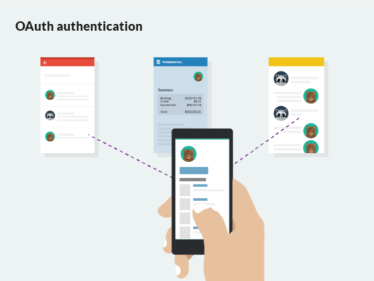
This course is intended for junior developers with some experience using APIs and web services.

Audience

Software Developers

Time Required

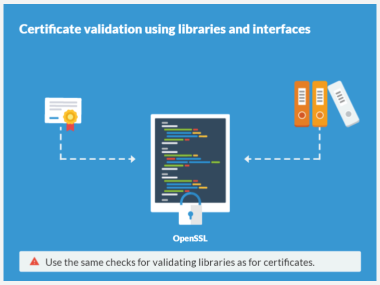
Tailored learning - 75 minutes total



OAuth authentication

Vulnerabilities in JSON files

Never build JSON dynamically, this increases the chances of XSS.



Certificate validation using libraries and interfaces

Use the same checks for validating libraries as for certificates.

Security Compass

API101 - DEFENDING WEB APIS

Course Outline

1. Authentication and authorization

- About RESTful Web APIs
- Public and Private APIs
- Security weaknesses in Web APIs
- Common vulnerability
- Guidelines for performing checks on RESTful web APIs
- Private vulnerability
- Defense - IdP
- Public vulnerability
- Defense - API keys
- Guidelines for using API keys
- Design safe RESTful services
- The client side
- The server side
- Token-based authentication
- How token-based authentication works
- JSON web tokens
- Components of JSON web token
- Best practices for using JWT
- OAuth authentication
- API Key vs. OAuth vs. JWT

2. Input validation and output encoding

- Improper input validation
- What is external user input?
- Validate all forms of input
- Best practices for validating input and incoming data for REST services
- Missing or incorrect XML validation
- Validate all XML input
- Vulnerabilities in JSON files
- Validating JSON using a JSON schema
- JSON schema examples
- Insecure deserialization
- XML output encoding
- JSON output encoding
- Neutralizing input during web page generation
- Best practices for escaping untrusted data

3. Secure communication

- Unencrypted transmission of sensitive data
- Vulnerabilities in TLS/SSL communication
- Best practices for securing TLS/SSL communication in Web APIs
- About ciphers
- Best practices for choosing a secure cipher
- List of ciphers
- SSL stripping
- Use HTTP Strict Transport Security
- Include the preload directive
- Certificate validation
- Best practices for validating certificates and chains of trust
- Certificate validation using libraries and interfaces
- Additional measures for certificate validation

4. Reducing attack surface

- Reducing the attack surface of Web APIs
- DoS
- Account lockout and authentication throttling
- DDoS
- Rate limiting
- CSRF attack
- Anti-CSRF tokens
- Best practices for using anti-CSRF tokens
- Cross-Origin Resource Sharing
- Best practices for securing CORS
- Debug data, extra files, and storage
- Disable and remove debug capabilities and data
- Information exposure
- Best practices for error and exception handling
- HTTP status codes
- Guidelines for using status codes
- Logging

4. cont.

- Best practices for logging critical security events
- Information leaks
- Best practices for logging confidential data

Security Compass