NOD201 - DEFENDING NODE.JS

Course Learning Objectives

Defending Node is comprised of four modules that cover common risks and best practices for secure coding. In the first module, we begin with developing safe JavaScript code using linters, clarifying code with promises, and choosing safe third-party modules. In the second module, we cover the most common attacks that target web-facing software, using HTTP headers and TLS encryption, and hardening your application against DoS and CSRF attacks. In module three, we explore safe ways to deal with different types of data by securing sensitive details and preventing injection attacks in user-supplied data. Finally, in the last module, we explore how you can secure Node in a production environment.

Description

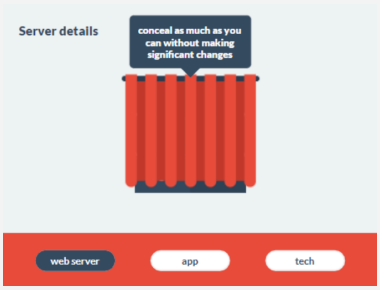
This course is designed for Node and Web developers who have some familiarity with Web application security. Node is one of the most commonly used open-source Web technologies for building scalable web applications. For this reason, it's important to understand its security risks and how to implement defensive coding techniques and configurations.

Audience

Node.js Developers
Web Developers

Time Required

Tailored learning - 60 minutes total (approx.)



NOD201 - DEFENDING NODE.JS

Course Outline

1. Following Safe Coding Practices

- JavaScript in Node
- Safe JavaScript
- Checking code with ESLint
- Protecting the event loop
- Best practices for using Node safely
- Asynchronous promises - callback hell
- Asynchronous promises - solution
- Malicious packages
- Security exploits
- How malicious packages work
- Protecting packages

2. Defending Against Web Attacks

- Web attacks
- Using Helmet
- HTTP headers
- Choosing headers
- Transport Layer Security
- Best practices for configuring TLS in Node
- HTTP Strict Transport Security
- Denial of Service attacks
- Configure a maximum request size
- Use the right server timeouts
- Avoid unbounded work
- Avoid unsafe regular expressions
- Use a rate-limiting module to restrict excessive requests
- Add extra protection to authentication pages
- Cross-Site Request Forgery attacks
- CSRF defenses
- Secure cookies
- Best practices for securing cookies

3. Dealing with Data

- Handling secrets
- Environment variables
- Configuration files
- Cloud-based secrets
- Sensitive server information
- Server details
- Error messages
- Timing information
- Injection attacks
- Untrusted data
- Sanitization
- Injection types
- SQL injection
- HTML injection
- Code injection
- Prototype pollution
- File path injection

4. Secure Node Deployment

- Node in production
- Version policy
- Node versioning
- LTS releases
- Versioning guidelines
- Experimental features
- Locating experimental features
- The Node process
- Privileged ports
- Handling privileged ports
- Monitoring package vulnerabilities
- Monitoring application logs

SecurityCompass