# Node JS Secure Coding

## 3-Day Course

## Course Syllabus

# Node JS secure coding
# 3-day course

## Course description

Secure programming is the best defense against hackers. This multilayered hands-on course will demonstrate live real time hacking methods , analyze the code deficiency that enabled the attack and most importantly, teach how to prevent such vulnerabilities by adopting secure coding best practices in order to bullet-proof your Node JS application.

The methodology of the Cycle of knowledge is as follows: Understand, Identify, Prevent. This methodology presents the student with analytical tools to keep a deeper understanding of coding vulnerabilities and implement security countermeasures in different areas of the software development lifecycle. The hands on labs will enable the student to get a firsthand experience of the Hackers world and what could be done to stop him. Using sound programming techniques and best practices shown in this course, you will be able to produce high-quality code that stands up to attack.

The course covers major security principles in the Node JS applications, programming vulnerabilities, and specific security issues in Node JS web applications.

## Target audience

Members of the software development team:

- Node JS developers
- Designers & architects.

## Prerequisites

Before attending this course, students should be familiar with:

- Basic knowledge of Node JS

# Course topics

Day 1:

## Authentication

- What is authentication
- Store password securely
- Hashing
- Brute force
- Dictionary attack
- Anti-automation
- CAPTCHA
- Account lockout
- User enumeration
- Revealing too much information
- **Lab: Authentication bypassing and enforcement**

## Authorization

- Client side authorization
- Forceful browsing
- UI based security
- Implement authorization using middleware
- Parameter tampering
- Insecure direct object reference
- File authorization
- URL authorization
- **Lab: Impersonation and authorization**

## Injections and Input Validation

- OS command injection
- SQL Injection
- Prepared statement
- Mongo DB (NoSQL) Injection
- Xpath injection
- LDAP injection
- Data type conversion
- Black list

- White list
- Regular expression
- **Lab: Data manipulation via SQL Injection and mitigations**

Day 2:

## Output Encoding

- Reflected / Stored Cross site scripting
- XSS threats
- Encoding types
- XSS prevention cheat sheet
- Node JS Encoding libraries
- Swig template engine
- **Lab: Identity theft via XSS and mitigations**

## Browser Manipulation

- Cross Site Request Forgery (CSRF)
- Anti CSRF token
- Open redirect
- Clickjacking
- Auto complete
- Browser’s cache
- Session management
- Cookie's properties
- Session fixation
- Helmet middleware
- **Lab: Use security headers against browser manipulation attacks**

## File Handling

- Directory traversal
- Canonicalization
- File extension handling
- Filenames threats
- Sanity filename with fs.realpathSync
- Directory listing
- **Lab: Protect against malicious file upload**

Day 3:

## Data Confidentiality & Integrity

- Homemade algorithm
- Insecure communication
- Secure traffic enforcement
- Insecure storage
- Symmetric encryption
- A-Symmetric encryption
- Java Cryptography Architecture (JCA)
- Hash functions
- Digital signatures
- **Lab: Integrity and reliability**

## Error Handling

- Information disclosure
- Exceptions and stack trace
- Default error pages
- Customize error pages using express
- **Lab: Sensitive information leakage**

## Security Logging

- Logging technologies
- Events you should log
- Events you should not log
- **Lab: Log spoofing and secure logging**

## Business Logic

- Logical attacks
- Flow bypassing
- Replay attacks
- Abuse of functionality
- **Lab: business logic flaws**