# Beginner's Guide to Basic Web Application Security

This project is a hands-on introduction to fundamental web security concepts using Node.js. The goal is to demonstrate how to build a simple multi-port server, test it for open ports using Nmap, and implement a basic logging system with Winston.

## Table of Contents

* [Project Overview](https://www.google.com/search?q=%23project-overview)
* [Features](https://www.google.com/search?q=%23features)
* [Prerequisites](https://www.google.com/search?q=%23prerequisites)
* [Installation & Setup](https://www.google.com/search?q=%23installation--setup)
* [Usage](https://www.google.com/search?q=%23usage)
  + [Running the Server](https://www.google.com/search?q=%231-running-the-server)
  + [Testing with Nmap](https://www.google.com/search?q=%232-testing-with-nmap)
  + [Verifying Logs](https://www.google.com/search?q=%233-verifying-logs)
* [Security Checklist](https://www.google.com/search?q=%23security-checklist)

## Project Overview

This repository provides the code and documentation for a beginner-level web security exercise. It covers three core topics:

1. **Basic Penetration Testing:** Creating a server with multiple open network ports and using **Nmap** to discover them.
2. **Application Logging:** Implementing a robust logging system using the **Winston** library to track application events.
3. **Security Best Practices:** Reviewing a fundamental security checklist applicable to any web project.

## Features

* Simple multi-port web server built with **Express.js**.
* Demonstrates network scanning with the **Nmap** tool.
* Professional-grade logging to both the console and a file using **Winston**.
* Clear, step-by-step instructions for setup and testing.

## Prerequisites

Before you begin, ensure you have the following software installed on your computer:

* **Node.js and npm:** [Download & Install Node.js](https://nodejs.org/)
* **Nmap:** [Download & Install Nmap](https://nmap.org/download.html)

## Installation & Setup

Follow these steps to get the project running on your local machine.

1. **Clone the repository:**  
   git clone [https://github.com/your-username/final-security-project.git](https://github.com/your-username/final-security-project.git)  
   cd final-security-project
2. **Initialize the Node.js Project:** This command creates a package.json file.  
   npm init -y
3. **Install Dependencies:** Install the required express and winston packages.  
   npm install express winston

## Usage

### 1. Running the Server

Start the application by running the server.js file from your terminal.

node server.js

You should see output confirming that all three servers have started.

info: Application is starting up...  
info: Server 1 is running and listening on http://localhost:3000  
info: Server 2 is running and listening on http://localhost:8080  
info: Server 3 is running and listening on http://localhost:9000

**Important:** Keep this terminal window open while you perform the tests.

### 2. Testing with Nmap

Open a **new, separate terminal window** to scan your running application.

nmap localhost

The output will confirm that the ports are open and visible on your network.

**Expected Nmap Output:**

PORT STATE SERVICE  
3000/tcp open ppp  
8080/tcp open http-proxy  
9000/tcp open cslistener

### 3. Verifying Logs

1. **Trigger a Log Event:** Open your web browser and navigate to http://localhost:3000.
2. **Check the Console:** The terminal where your server is running will display a new log message for your visit.
3. **Check the Log File:** A new file, security.log, has been created in your project directory. It contains a permanent JSON record of all application events.

## Security Checklist

This project demonstrates the "how-to" for a few items on a standard security checklist. For any real-world application, ensure you address all relevant points:

| **Category** | **Best Practice** |
| --- | --- |
| **Input & Output** | Validate All Inputs from users. |
| **Data Security** | Hash and Salt Passwords securely. |
| **Data Transmission** | Use HTTPS to encrypt all traffic. |
| **Access Control** | Enforce the Principle of Least Privilege. |
| **Error Handling** | Hide detailed error messages from users. |
| **Dependencies** | Regularly audit third-party code with npm audit. |