Cyber Security Internship

Week # 3-Documentation

Project Title:Final Phase: Application Security Testing and ReportingIntern: Syeda ayesha khan

Tool Stack: Node.js, Express.js, MongoDB, OWASP ZAP, Browser Developer Tools



**Week 3 – Penetration Testing, Network Scanning, Logging, Final Recommendations**

1. **Penetration Testing Results**

✅ SQL Injection (Retested)

Payload Used: ' OR '1'='1

Application returned: "Invalid characters in username"  
Successfully blocked.

✅ XSS Attempt (Retested)

Payload Used: <script>alert("XSS")</script>

Application returned: "Invalid characters in username"  
Input validation effectively blocked script injection.

**2. Nmap Port Scanning**

Conducted a network scan on localhost using Nmap:

Command Used:

nmap localhost

**Scan Output:**

PORT STATE SERVICE

135/tcp open msrpc

445/tcp open microsoft-ds

5357/tcp open wsdapi

**Analysis:**

* Port 135 (MS-RPC): Should be blocked or restricted via firewall on production.
* Port 445 (SMB): Commonly exploited. Disable unless absolutely required.
* Port 5357 (WSDAPI): Should be accessible only within trusted internal networks.

**3. Logging Setup with Winston**

Implemented application-level logging to track security-related events.

Code Snippet:

const winston = require('winston');const logger = winston.createLogger({

transports: [

new winston.transports.Console(),

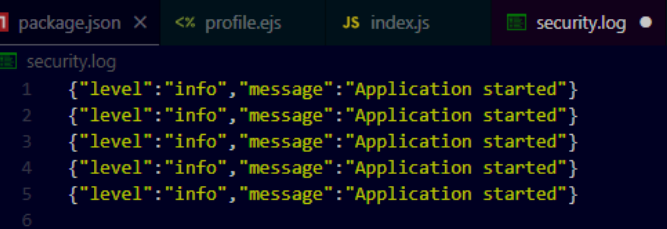
new winston.transports.File({ filename: 'security.log' })

]

});

logger.info('Application started');

✅ Result: Logs successfully written to security.log, confirming traceability and audit readiness.



1. **Security Best Practices Checklist**

| Security Practice | Status |
| --- | --- |
| Input validation using validator.js | ✅ Implemented |
| Secure HTTP headers with helmet.js | ✅ Implemented |
| Password hashing using bcrypt | ✅ Already Present |
| JWT for session/token-based authentication | ✅ Implemented |
| Application logging with winston | ✅ Implemented |
| Use of HTTPS for secure communication | Recommended for Production |
| Network port scanning and firewall review | ✅ Conducted |
| Removal of suspicious comments and debug info | ✅ Cleaned Up |

**✅ Conclusion**

Over the span of three weeks, I successfully conducted a full-scale security assessment and improvement of a Node.js-based user management web application. The transition from identifying vulnerabilities to securing the platform involved:

* Practical use of automated and manual testing tools
* Implementation of modern security libraries
* Adoption of industry-standard best practices

The application is now significantly more resilient to common web attacks and adheres to OWASP guidelines. Remaining production-level tasks (like HTTPS enforcement and deployment hardening) are also noted for final implementation.