**Cybersecurity Internship Report**

**Intern Name:** Aiman Asif  
**Project Title:** Strengthening Security Measures for a Web Application  
**Submitted to:** Faizan Khan  
**Date:** 29May, 2025

**Week 3: Advanced Security and Final Reporting**

**1. Basic Penetration Testing (Manual Simulations)**

**Objective:**  
To simulate real-world attacks manually to test how the application handles malicious behaviors.

**Steps Performed:**

* **Input Attacks:**
  + Repeated XSS payloads (like <script>alert('XSS');</script>) in login, signup, and profile forms.
  + Submitted empty or malformed form data (e.g., blank emails or names, invalid password formats).
  + ✅ **Secure behavior observed**: Application rejected invalid inputs due to validation and sanitization added earlier.
* **Session Tampering via Browser DevTools:**
  + Opened **Developer Tools > Application > Cookies**.
  + Located session cookie (connect.sid).
  + Manually edited its value to random/invalid data (e.g., fake12345sidtoken).
  + Refreshed the page.
  + ✅ **Expected result achieved**: App redirected to the login page, indicating proper session invalidation.
* **Skipped Nmap:**
  + Nmap scanning was **optional** and was **not performed** due to time constraints.
  + Manual browser-based testing provided sufficient insight into vulnerabilities.

**2. Logging Security Events (Using Winston)**

**Objective:**  
To track critical security events like application start, errors, and route access to improve visibility and auditing.

**Tool Used:**

* [winston](https://www.npmjs.com/package/winston)

**Steps Taken:**

* Installed Winston via:

nginx

CopyEdit

npm install winston

* Configured a logger in app.js:

js

CopyEdit

const winston = require('winston');

const logger = winston.createLogger({

transports: [

new winston.transports.Console(),

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

]

});

* Logged important events:
  + logger.info('Application started');
  + logger.warn('404 - Page not found');
  + logger.error('Error during port listening: ...');
* Verified logs in security.log file for:
  + App start
  + 404 errors
  + Manual logs for route visits

**3. Security Checklist**

| **#** | **Security Measure** | **Status** | **Notes** |
| --- | --- | --- | --- |
| 1 | **Input Validation** | ✅ Implemented | Using validator library for email and other checks |
| 2 | **Password Hashing** | ✅ Secure | Using bcrypt (already implemented in Week 1) |
| 3 | **Secure Transmission** | ✅ Helmet used | helmet middleware added for HTTP header protection |
| 4 | **Session Handling** | ✅ Working | Session-based login, with cookie tampering tests passed |
| 5 | **Logging Events** | ✅ Winston Setup | Logs recorded in security.log |
| 6 | **Penetration Testing** | ✅ Manual Testing Done | XSS, SQLi, and session tampering simulated |
| 7 | **Nmap Scan** | ❌ Skipped | Optional, not done due to time constraints |

**Summary**

In Week 3, I focused on:

* Simulating real-world attacks manually
* Validating session and cookie handling
* Implementing event logging using Winston
* Creating a checklist of critical security best practices

This final week helped ensure the application not only prevents known vulnerabilities (like XSS and SQLi) but also **logs suspicious behavior**, handles **session hijacking attempts**, and follows **basic security hygiene**.