Cybersecurity Interns Task (Weeks 4–6)

Deadline: July 24, 2025

Week 4: Advanced Threat Detection & Web Security Enhancements

Goal:

Implement advanced security measures, detect threats in real-time, and secure API endpoints.

Tasks:

1. Intrusion Detection & Monitoring

- Set up real-time monitoring using **Fail2Ban** or **OSSEC**.
- Configure alert systems for **multiple failed login attempts**.

2. API Security Hardening

- Apply **rate limiting** using `express-rate-limit` to prevent brute-force attacks.
- Properly configure **CORS** to restrict unauthorized access.
- Secure APIs using **API keys** or **OAuth authentication**.

3. Security Headers & CSP Implementation

- Implement **Content Security Policy (CSP)** to prevent script injections.
- Enforce HTTPS using **Strict-Transport-Security (HSTS)** headers.

Deliverables:

- Secured API with **rate-limiting** and **authentication** mechanisms.
- Implemented **security headers** with proper configuration.
- **GitHub repository** containing code updates and a detailed `README.md`.

Week 5: Ethical Hacking & Exploiting Vulnerabilities

Goal:

Learn ethical hacking techniques, exploit vulnerabilities in a test environment, and enhance application security.

Tasks:

1. Ethical Hacking Basics

- Use **Kali Linux** or any preferred penetration testing toolkit.
- Conduct **reconnaissance** on a **test web application**.

2. SQL Injection & Exploitation

- Use **SQLMap** to identify SQL injection vulnerabilities.
- Prevent SQLi by applying **prepared statements** in your backend code.

3. Cross-Site Request Forgery (CSRF) Protection

- Implement CSRF protection using the `csurf` middleware in **Node.js**.

- Test CSRF vulnerabilities using **Burp Suite**.

Deliverables:

- **Ethical hacking report** with details of vulnerabilities found.
- Security fixes for **SQLi** and **CSRF** implemented in the code.
- Updated **GitHub repository** with security improvements and documentation.

Week 6: Advanced Security Audits & Final Deployment Security

�� Goal:

Conduct advanced security audits, ensure compliance with industry standards, and prepare the application for secure deployment.

Tasks:

1. Security Audits & Compliance

- Conduct security audits using:
    - **OWASP ZAP**
    - **Nikto**
    - **Lynis**
- Check compliance with **OWASP Top 10** best practices.

2. Secure Deployment Practices

- Enable **automatic security updates** and **dependency scanning**.
- Follow **Docker security best practices**, including scanning container images for vulnerabilities.

3. Final Penetration Testing

- Perform a comprehensive penetration test using tools like **Burp Suite** or **Metasploit**.
- Document vulnerabilities, test results, and applied security improvements.

Deliverables:

- Final **security audit report**.
- Fully **secured and deployed application**.
- **GitHub repository** with all applied security fixes and updated documentation.
- **4–5 minute video recording** of the project with voiceover explaining the security implementation.

Bonus Challenge (Optional, for Excellence):

- Implement **Zero Trust Security principles** for user authentication and resource access. ●

Deploy a **Web Application Firewall (WAF)** for added protection.

- Simulate **Social Engineering Attacks** (e.g., phishing awareness training) and document findings.