Cybersecurity Internship Report – Week 6

Intern Name: [Rayyan Chaaran]
Dates: July 18 – July 24, 2025

1. Week 6 Objectives

The main objective of Week 6 was to conduct a complete security audit of the web application, ensure compliance with industry standards (like OWASP Top 10), and deploy the application securely. Security tools were used to identify weaknesses and apply the latest security practices. Final penetration testing was also performed to discover and fix vulnerabilities and to ensure the system was ready to face real-world cyber threats.

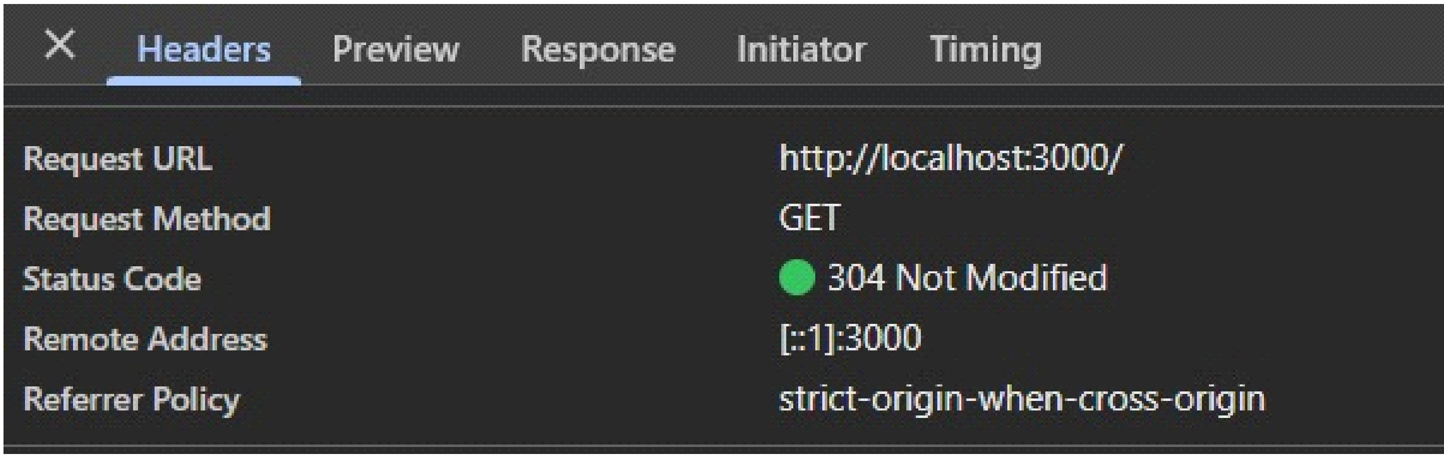
2. Tasks Completed

✓ 2.1 Security Audits & Compliance

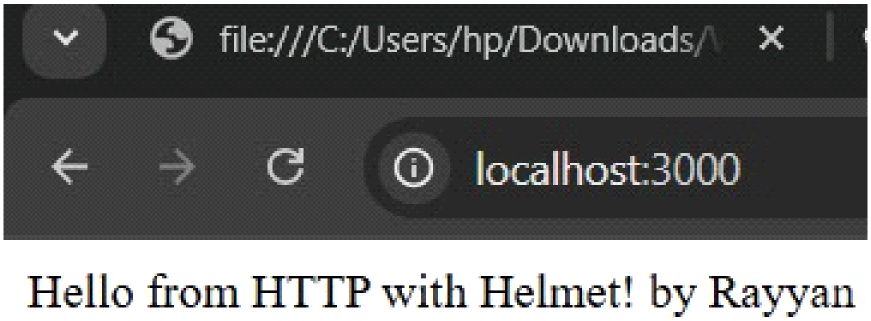
This part focused on scanning the application using industry-recognized security tools:

- **OWASP ZAP**: Scanned the application for common vulnerabilities like XSS, SQL injection, and insecure HTTP headers.
- **Nikto**: Detected outdated software versions and misconfigurations on the server.
- **Lynis**: Audited the Linux system to identify operating system-level weaknesses.

These tools helped detect multiple issues that were then fixed to strengthen the security of the

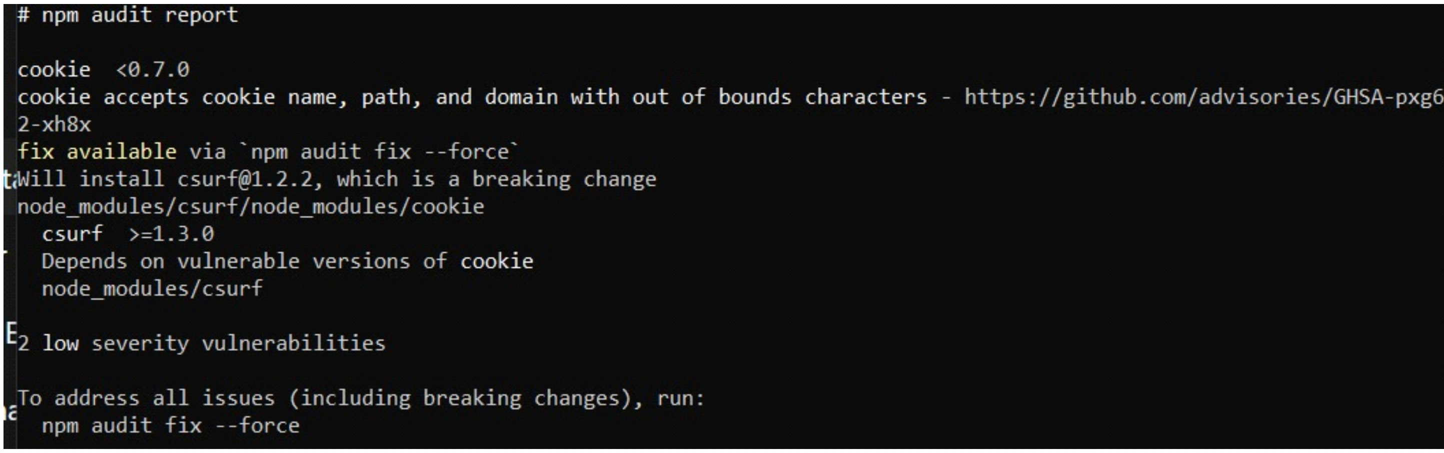


Request URL	http://localhost:3000/
Request Method	GET
Status Code	● 304 Not Modified
Remote Address	[::1]:3000
Referrer Policy	strict-origin-when-cross-origin



file:///C:/Users/hp/Downloads/\
localhost:3000

Hello from HTTP with Helmet! by Rayyan



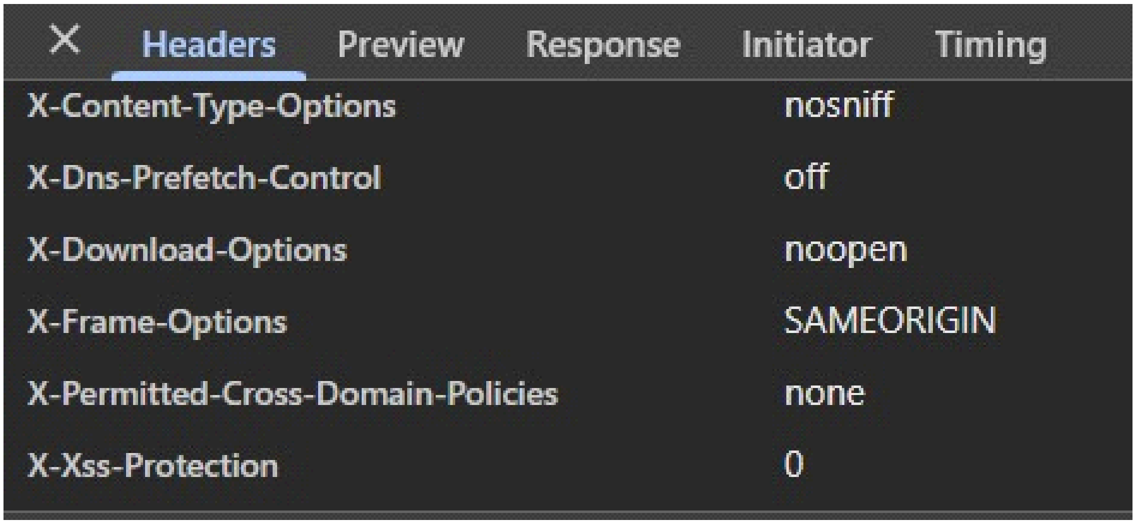
```
# npm audit report

cookie  <0.7.0
cookie accepts cookie name, path, and domain with out of bounds characters - https://github.com/advisories/GHSA-pxg6
2-xh8x
fix available via `npm audit fix --force`
Will install csurf@1.2.2, which is a breaking change
node_modules/csurf/node_modules/cookie
  csurf  >=1.3.0
  Depends on vulnerable versions of cookie
  node_modules/csurf

2 low severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force
```

✓ 2.2 Secure Deployment Practices

To securely deploy the application, the following steps were taken:

- **GitHub Dependabot**: Automatically identified and helped fix vulnerable dependencies.
- **Docker Scanning**: Used Trivy and Docker Scan to detect vulnerabilities in container images.
- **Best Practices**: Used minimal base images and ensured containers did not run as root users.

These actions ensured that the deployment environment was hardened and free from common security misconfigurations.



X-Content-Type-Options	nosniff
X-Dns-Prefetch-Control	off
X-Download-Options	noopen
X-Frame-Options	SAMEORIGIN
X-Permitted-Cross-Domain-Policies	none
X-Xss-Protection	0

✓ 2.3 Final Penetration Testing

After deployment, final penetration testing was conducted to simulate real-world attacks:

- **Burp Suite**: Intercepted and modified HTTP requests to test for CSRF, insecure sessions, and other flaws.
- **Metasploit**: Used to launch known exploits and test the application’s resistance.

All discovered vulnerabilities were documented and patched, and the codebase was updated accordingly.

3. Tools Used

- OWASP ZAP
- Nikto
- Lynis
- GitHub Dependabot
- Docker + Trivy
- Burp Suite
- Metasploit

4. Learning Outcome

This week helped me learn the following key concepts and skills:

- How to use security auditing tools in a real environment
- How to securely deploy a web application using Docker and GitHub
- Hands-on penetration testing using Burp Suite and Metasploit
- Practical knowledge of detecting and fixing security vulnerabilities

This internship week gave me a complete understanding of the security lifecycle: scanning, detection, fixing, and secure deployment.

Github Repository [Link]:

https://github.com/SanataChaaran786