Cyber security report

Team Members

- Abdelwahab Mohamed 2305072
- Belal Magdy 2305063
- Gamal Mahmoud 2305068

Key Findings:

- Critical Vulnerabilities Identified:

1. Lack of access control in admin functionality.
2. Absence of rate-limiting, allowing brute force attacks.
3. Cross-Site Scripting (XSS) in product search input.

Impact: Exploitation of these vulnerabilities can result in unauthorized access, data theft, and user session hijacking.

Recommendations:

- Implement rate-limiting and account lockout mechanisms.
- Sanitize and validate user inputs to prevent XSS.
- Secure hidden paths and admin functionality with robust access controls.

Scope and Methodology

Scope:
Testing was limited to the OWASP Juice Shop web application, including APIs and key user workflows.

Testing Approach:

- **Type:** Black-box testing.
- **Tools Used:** ZAP

Vulnerability Findings

1. Enumeration to Find Admin Path

- **Description:** The application lacks proper access control, exposing admin paths through URL guessing.
- **Risk/Impact:** Enables attackers to target privileged areas for further exploitation.

- **Remediation:** Implement obfuscation for admin paths and enforce authentication checks.

2. Brute Force on Admin Credentials

- **Description:** Using Hydra, attackers can brute force admin credentials due to the absence of rate-limiting.
- **Risk/Impact:** Full control over the application is granted to the attacker.
- **Evidence:** Successful login via brute force attack.
- **Remediation:** Add rate-limiting and account lockouts.

3. XSS in Product Search

- **Description:** Malicious scripts can be executed via unsanitized inputs in the product search.
- **Risk/Impact:** Results in theft of sensitive data and session hijacking.
- **Evidence:** Demonstrated script execution and session cookie theft.
- **Remediation:** Use input sanitization and output encoding.

Conclusion

Security Posture:
The OWASP Juice Shop exhibits significant security weaknesses. The overall risk level is high, with immediate action required.

Next Steps:

- Prioritize fixing identified vulnerabilities.
- Conduct further security assessments post-remediation.
- Train developers on secure coding practices.

Attack videos link

https://drive.google.com/drive/folders/14jruTubeUrowND2aI1VVnQLCJsUPRkEv?usp=drive_link