CyS Project - Team 31-B

Summary

This penetration testing report provides an overview of the security vulnerabilities identified in the target system. The primary focus of this assessment was to identify and exploit potential security weaknesses that could compromise the confidentiality, integrity, and availability of the system. The testing revealed three significant vulnerabilities, including improper access control and a cross-site scripting (XSS) vulnerability. This report outlines the findings, methodologies used, and recommendations to address the vulnerabilities.

Scope & Methods

Scope

- Target System: Juice Shop Web Application
- Objective: Identify and exploit vulnerabilities to demonstrate potential security risks.

Methods Used

1. **Firefox Developer Tools**: Used to inspect and analyze the web application's code and behavior.
2. **Kali Linux**: Employed for testing tools and scripts.
3. **NPM (Node Package Manager)**: Used for dependency analysis and testing relevant JavaScript components.

Vulnerability Findings

1. Hidden Scoreboard Page

- **Description**: A hidden scoreboard page was discovered through the `main.js` file found in the browser's developer tools.
- **Details**: The JavaScript file contained a path reference to `/score-board`, which led to sensitive application functionality.
- **Impact**: Unauthorized access to sensitive application pages could allow attackers to gain insights into the system's structure.

2. Admin Section Exposure

- **Description**: Similar to the hidden scoreboard page, the admin section path was revealed by inspecting `main.js`.
- **Details**: The admin page path allowed unauthorized access to administrative functionalities.
- **Impact**: Potential compromise of administrative controls and escalation of privileges.

3. Admin Password Exposure

- **Description**: The admin password was exposed through analysis of `main.js` in developer tools.

- **Details**: Weak obfuscation or poor handling of sensitive information in the JavaScript file led to this vulnerability.
- **Impact**: Direct compromise of the administrator account.

4. Cross-Site Scripting (XSS)

- **Description**: A reflected XSS vulnerability was found in the search bar functionality of the application.
- **Details**: Searching for terms manipulated the URL query parameters. By injecting the following code, cookies were accessed:

```
<iframe src="javascript: alert(document.cookie)"></iframe>
```

- **Impact**: Exploitation of this vulnerability could result in session hijacking or theft of sensitive user data.

Exploitation and Attack Simulation

Hidden Scoreboard Page

- Accessed the `/score-board` page directly via the browser after identifying the path in `main.js`.

Admin Section

- Navigated to the admin section path identified through developer tools to simulate unauthorized access.

Admin Password

- Extracted the admin password from the `main.js` file and successfully logged into the admin panel to demonstrate account compromise.

XSS Exploitation

- Injected malicious JavaScript code in the search bar, leading to an alert displaying cookies. This demonstrated the potential for session hijacking and further exploitation.

Conclusion

The penetration test revealed critical vulnerabilities in the target system:

1. Lack of proper access control for sensitive application paths.
2. Insecure handling of sensitive information in JavaScript files.
3. Inadequate input validation leading to XSS.

Recommendations

1. Implement access controls to restrict unauthorized access to sensitive pages such as the scoreboard and admin panel.
2. Avoid exposing sensitive information, such as passwords, in client-side JavaScript files.
3. Sanitize and validate all user inputs to prevent XSS and similar attacks.
4. Conduct regular security assessments and code reviews to identify and mitigate vulnerabilities.

By addressing these issues, the overall security posture of the application can be significantly improved.