Mitigation Report – OWASP Juice Shop

Note on Lab Setup

In this lab we tried to set up Nginx with ModSecurity Web Application Firewall (WAF) to block attacks on Juice Shop. The Juice Shop container was running fine, but the Nginx WAF container was not connecting properly. Because of this, we were not able to complete the mitigation practically. So in this report we explain the mitigation steps in theory.

Security Control Selection

The main attacks we performed were:

- Direct access to administrator page.
- Weak password login with admin123.
- SQL Injection using ' OR 1=1--.

To stop these attacks, the following security controls should be applied:

- Web Application Firewall (WAF): Use Nginx with ModSecurity to block SQL Injection and XSS payloads.
- Access Control: Restrict administrator pages so that only logged-in admin users can open them.
- Strong Authentication: Enforce strong password rules and disable weak default passwords.
- Secure Coding Practices: Use parameterized queries to stop SQL Injection.
- Input Validation: Sanitize and escape user input to prevent XSS.

Configuration Changes

If the WAF was working, the following changes would be made:

- In Nginx config, enable ModSecurity with OWASP Core Rule Set (CRS).
- Add rules to block SQL keywords like OR 1=1 and comment symbols --.
- Add rules to block <script> tags and suspicious HTML input.
- Redirect all traffic through Nginx WAF before reaching Juice Shop.
- In Juice Shop code, replace string concatenation with parameterized queries.
- In authentication system, enforce password length minimum 8 characters, with mix of letters, numbers, and symbols.

Patches and Recommendations

- Update Juice Shop to latest version where many vulnerabilities are already patched.
- Apply secure password policy and force password change for admin account.
- Use HTTPS for secure communication.
- Store secrets and credentials in environment variables, not hardcoded in code.
- Regularly update Docker images to latest security patches.

Validation (Theoretical)

If the WAF was working correctly, the validation would be:

- Try SQL Injection ' OR 1=1-- → WAF should block and show error page.
- Try weak password admin123 → System should reject because password policy does not allow weak passwords.
- Try opening administrator page directly → Access should be denied unless logged in as admin.

Before mitigation: all attacks were successful. After mitigation: attacks should be blocked and not successful.

Conclusion

In this lab, mitigation could not be tested practically because of WAF setup issue. But theoretically, the solution is clear. By using Nginx with ModSecurity, applying strong authentication, fixing access control, and coding securely, the attacks we performed can be stopped. This shows the importance of combining WAF with secure development practices to protect web applications like Juice Shop.