Broken Authentication Security Lab Project

Using Burp Suite for Authentication & Session Testing

Author:
Date:

1. Executive Summary

Broken Authentication is one of the OWASP Top 10 vulnerabilities. It occurs when applications improperly implement login, session, or password reset mechanisms, allowing attackers to compromise accounts. In this lab, we tested an intentionally vulnerable OWASP web application at 192.168.29.186 using Burp Suite. The goal was to identify authentication flaws, demonstrate them safely in a lab, and provide security recommendations.

2. Lab Setup

- **Target VM:** OWASP DVWA / BWA / Juice Shop at 192.168.29.186 - **Attacker VM:** Kali Linux with Burp Suite Community Edition - **Browser:** Firefox configured with Burp proxy (127.0.0.1:8080) - **Test Accounts:** victim:test123, attacker:hacker123 (lab only) - **Network:** Host-only adapter (192.168.29.0/24)

3. Step-by-Step Testing Procedures

3.1 Weak Login Protections

1. Intercepted login requests with Burp Proxy while attempting valid/invalid credentials. 2. Sent captured requests to Burp Repeater for replay with different usernames/passwords. 3. Observed verbose error messages (e.g., 'invalid username' vs. 'invalid password'). 4. Verified lack of account lockout or rate limiting by repeating attempts.

3.2 Cookie Security

1. Logged into the application and captured Set-Cookie headers in Burp. 2. Inspected flags: Secure, HttpOnly, SameSite. 3. Modified cookie values in Burp Repeater to test if the server accepts tampered cookies. 4. Checked if cookies had long expiry times or predictable values.

3.3 Session Fixation

1. Captured session cookie before login. 2. Logged in and checked whether the session cookie rotated to a new value. 3. Logged out and attempted to reuse the old cookie in Burp Repeater. 4. Verified whether the server continued to accept the old session.

3.4 Forgot Password Flow

1. Initiated a password reset request and intercepted the response in Burp. 2. Inspected reset token format for predictability (e.g., sequential IDs, Base64 encoding). 3. Attempted to reuse the same reset token after a password change. 4. Checked whether tokens expired after use or time.

3.5 Authorization Bypass

1. Accessed authenticated endpoints directly by removing session cookies in Burp Repeater. 2. Modified user IDs in parameters to test for Insecure Direct Object References (IDOR). 3. Observed whether the server granted access without proper authorization checks.

4. Evidence Collection

- Saved Burp project file (.burp) containing all requests and responses. - Documented findings with request/response pairs in Burp Repeater. - Recorded cookie attributes, tokens, and error messages for analysis. - No screenshots included, but placeholders exist for documentation.

5. Findings Table

ID	Category	Severity	Description	Evidence	Remediation
BA-01	Weak Login	High	Unlimited login attempts	Burp Repeater logs	Implement rate limiting & lockout
BA-02	Cookie	Medium	Missing HttpOnly flag	Set-Cookie response	Add HttpOnly, Secure, SameSite flags
BA-03	Session	Medium	No session rotation on login	Cookie comparison	Rotate session IDs at login/logout
BA-04	Reset Flow	High	Token reuse possible	Reset link evidence	Enforce single-use tokens with expiry
BA-05	Authorization	High	Direct access without login	Burp Repeater requests	Apply server-side access control

6. Mitigation & Recommendations

- Enforce account lockout, CAPTCHA, and MFA for login protection. - Secure cookies with Secure, HttpOnly, and SameSite flags. - Implement session rotation after login and invalidation after logout. - Use cryptographically strong, single-use reset tokens with expiry. - Enforce strict server-side authorization checks.

7. Conclusion

This project demonstrated common Broken Authentication vulnerabilities using Burp Suite against a vulnerable OWASP target. The tests highlighted risks in login, session, cookie, and password reset mechanisms. By applying the recommended mitigations, organizations can significantly reduce the risk of account compromise and unauthorized access.