01-Web-Application-Security-Checklist

Web Application Security Testing Checklist

A comprehensive checklist for manual and tool-assisted web application penetration testing, aligned with the OWASP Web Security Testing Guide (WSTG) and OWASP ASVS.

---

1. Reconnaissance & Information Gathering

- [ ] Enumerate subdomains (passive: crt.sh, Amass, Shodan; active: DNS brute force)
- [ ] Identify technology stack (Wappalyzer, HTTP headers, error pages)
- [ ] Review robots.txt, sitemap.xml, security.txt
- [ ] Search for exposed source control (.git, .svn), backup files, and config files
- [ ] Google/Shodan dorking for exposed panels, credentials, or documentation
- [ ] Check for exposed API documentation (Swagger/OpenAPI, GraphQL introspection)
- [ ] Identify third-party dependencies and known CVEs (JS libraries, CMS versions)
- [ ] Map the application: crawl, spider, and manually walk through every role/feature

2. Configuration & Deployment Management

- [ ] Test for default/sample content, admin panels, and debug endpoints
- [ ] Verify HTTP security headers (CSP, HSTS, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Permissions-Policy)
- [ ] Check TLS configuration (protocol versions, cipher suites, certificate validity)
- [ ] Test HTTP methods allowed (TRACE, PUT, DELETE, OPTIONS)
- [ ] Check for verbose error messages / stack traces leaking internals
- [ ] Review CORS configuration for overly permissive origins
- [ ] Check for directory listing and unprotected admin interfaces
- [ ] Validate cloud storage bucket permissions (S3, Azure Blob, GCS) if in scope

3. Identity & Authentication

- [ ] Test username enumeration (login, registration, password reset)
- [ ] Test for weak lockout/rate-limiting on login and OTP endpoints
- [ ] Test password policy (length, complexity, breached password checks)
- [ ] Test "remember me" and persistent session implementation
- [ ] Test password reset flow (token predictability, expiry, reuse, host header injection)
- [ ] Test multi-factor authentication (bypass, brute force, backup codes)
- [ ] Test for credential stuffing / brute-force protections
- [ ] Review autocomplete and caching of sensitive login fields

4. Session Management

- [ ] Verify session tokens are random, high-entropy, and not predictable
- [ ] Test session fixation
- [ ] Test session expiration (idle timeout, absolute timeout, logout invalidation)
- [ ] Verify Secure, HttpOnly, and SameSite cookie attributes
- [ ] Test for concurrent session handling and session invalidation on password change

- [ ] Test CSRF protections on state-changing requests
- [ ] Check for session tokens leaking in URLs, logs, or Referer headers

5. Access Control

- [ ] Test for horizontal privilege escalation (IDOR — access other users' data)
- [ ] Test for vertical privilege escalation (access admin functions as a low-priv user)
- [ ] Test direct object references (sequential/guessable IDs)
- [ ] Test forced browsing to restricted URLs/functions
- [ ] Test for missing function-level access control (hidden UI vs. server-side enforcement)
- [ ] Test multi-tenant data isolation (if applicable)

6. Input Validation & Injection

- [ ] SQL injection (error-based, blind, time-based, out-of-band) on all parameters
- [ ] NoSQL injection
- [ ] Command injection / OS command execution
- [ ] LDAP injection
- [ ] XML External Entity (XXE) injection
- [ ] Server-Side Template Injection (SSTI)
- [ ] Reflected, stored, and DOM-based Cross-Site Scripting (XSS)
- [ ] HTML/CSS injection
- [ ] Server-Side Request Forgery (SSRF), including cloud metadata endpoint access
- [ ] Path traversal / local file inclusion / remote file inclusion
- [ ] HTTP request smuggling / header injection / CRLF injection
- [ ] Mass assignment / parameter pollution

7. File Upload & File Handling

- [ ] Test file type/extension validation bypass (magic bytes, double extensions)
- [ ] Test for unrestricted file upload leading to code execution
- [ ] Test file size limits and denial-of-service via large files
- [ ] Test for path traversal in filenames
- [ ] Verify uploaded files are served from a non-executable, isolated location
- [ ] Test malware/AV scanning if implemented

8. Business Logic

- [ ] Test workflow bypass (skipping steps in multi-step processes)
- [ ] Test for race conditions (e.g., coupon redemption, funds transfer, inventory)
- [ ] Test negative values / integer overflow in quantity or price fields
- [ ] Test for abuse of promotional codes, referral systems, or free trials
- [ ] Test rate limiting on sensitive business functions

9. Client-Side Security

- [ ] Review JavaScript for hardcoded secrets/API keys
- [ ] Test DOM-based vulnerabilities (postMessage, innerHTML sinks)
- [ ] Check clickjacking protections (X-Frame-Options / frame-ancestors)
- [ ] Review local storage / session storage for sensitive data exposure
- [ ] Test Content Security Policy effectiveness (not just presence)

10. Cryptography & Data Protection

- [ ] Verify sensitive data is encrypted in transit (TLS everywhere) and at rest
- [ ] Check for use of weak/broken algorithms (MD5, SHA1, ECB mode)
- [ ] Verify secrets are not stored in plaintext (config files, source, DB)
- [ ] Test for sensitive data exposure in logs, URLs, or error messages
- [ ] Review PII handling against applicable regulations (GDPR, CCPA, etc.)

11. Denial of Service Resilience (if in scope)

- [ ] Test for resource-intensive endpoints (regex DoS, large payloads, unpaginated queries)
- [ ] Test rate limiting / throttling on public endpoints

12. Reporting Artifacts to Capture

- [ ] Screenshots/PoC for each finding
- [ ] Full request/response pairs (Burp/ZAP export)
- [ ] CVSS scoring per finding
- [ ] Remediation recommendations mapped to each issue

Use alongside the OWASP WSTG and ASVS for full technical test-case detail.