Application Penetration Testing Checklist

Information Gathering

Rendered Site Review

- Manually explore the site
- Spider/crawl for missed or hidden content
- Check the webserver metafiles for information leakage files that expose content, such as robots.txt, sitemap.xml, and .DS_Store
- Check the caches of major search engines for publicly accessible sites
- Check for differences in content based on user agent (e.g. mobile sites, accessing as a search engine crawler)
- Check webpage comments and metadata for information leakage
- Development Review

Development Review

- Check the web application framework
- Perform web application fingerprinting
- Identify technologies used
- Identify user roles
- Identify application entry points
- Identify client-side code
- Identify multiple versions/channels (e.g. web, mobile web, mobile app)

Hosting and Platform Review

- Identify web services
- Identify co-hosted and related applications
- Identify all hostnames and ports
- Identify third-party hosted content

Configuration Management

- Check for commonly used application and administrative URLs
- Check for old, backup, and unreferenced files
- Check HTTP methods supported and Cross Site Tracing (XST)
- Test file extensions handling
- Test RIA cross domain policy
- Test for security HTTP headers (e.g. CSP, X-Frame-Options, HSTS)
- Test for policies (e.g. Flash, Silverlight, robots)
- Check for sensitive data in client-side code (e.g. API keys, credentials)
- Secure Transmission
- Protocols and Encryption

Secure Transmission

Protocols and Encryption

- Check SSL version, algorithms, and key length
- Check for digital certificate validity (duration, signature, and CN)
- Check that credentials are only delivered over HTTPS
- Check that the login form is delivered over HTTPS
- Check that session tokens are only delivered over HTTPS
- Check if HTTP Strict Transport Security (HSTS) in use
- Test ability to forge requests
- Test web messaging (HTML5)
- Check CORS implementation (HTML5)

Web Services and REST

- Test for web service issues
- Test REST

Authentication

Application Password Functionality

- Test password quality rules
- Test remember me functionality
- Test password reset and/or recovery
- Test password change process
- Test CAPTCHA
- Test multi-factor authentication
- Test for logout functionality presence
- Test for default logins
- Test for out-of-channel notification of account lockouts and successful password changes
- Test for consistent authentication across applications with shared authentication schema/SSO and alternative channels
- Test for weak security question/answer

Additional Authentication Functionality

- Test for user enumeration
- Test for authentication bypass
- Test for brute force protection
- Test for credentials transported over an encrypted channel
- Test for cache management on HTTP (eg Pragma, Expires, Max-age)
- Test for user-accessible authentication history

Session Management

- Establish how session management is handled in the application (eg, tokens in cookies, token in URL)
- Check session tokens for cookie flags (httpOnly and secure)
- Check session cookie scope (path and domain)
- Check session cookie duration (expires and max-age)
- Check session termination after a maximum lifetime

- Check session termination after relative timeout
- Check session termination after logout
- Test to see if users can have multiple simultaneous sessions
- Test session cookies for randomness
- Confirm that new session tokens are issued on login, role change, and logout
- Test for consistent session management across applications with shared session management
- Test for session puzzling
- Test for CSRF and clickjacking

Authorization

- Test for path traversal
- Test for vertical access control problems (a.k.a. privilege escalation)
- Test for horizontal access control problems (between two users at the same privilege level)
- Test for missing authorization
- Test for insecure direct object references

Cryptography

- Check if data which should be encrypted is not
- Check for wrong algorithms usage depending on context
- Check for weak algorithms usage
- Check for proper use of salting
- Check for randomness functions

Data Validation

Injection

- Test for HTML Injection
- Test for SQL Injection
- Test for LDAP Injection
- Test for ORM Injection
- Test for XML Injection
- Test for XXE Injection
- Test for SSI Injection
- Test for XPath Injection
- Test for XQuery Injection
- Test for IMAP/SMTP Injection
- Test for Code Injection
- Test for Expression Language Injection
- Test for Command Injection
- Test for NoSQL injection

Other

- Test for Reflected Cross Site Scripting
- Test for Stored Cross Site Scripting
- Test for DOM based Cross Site Scripting
- Test for Cross Site Flashing
- Test for Overflow (Stack, Heap and Integer)
- Test for Format String
- Test for incubated vulnerabilities
- Test for HTTP Splitting/Smuggling
- Test for HTTP Verb Tampering
- Test for Open Redirection
- Test for Local File Inclusion
- Test for Remote File Inclusion
- Compare client-side and server-side validation rules
- Test for HTTP parameter pollution
- Test for auto-binding
- Test for Mass Assignment

- Test for NULL/Invalid Session Cookie
- Test for integrity of data
- Test for the Circumvention of Work Flows
- Test Defenses Against Application Mis-use
- Test That a Function or Feature Cannot Be Used Outside Of Limits
- Test for Process Timing
- Test for Web Storage SQL injection (HTML5)
- Check Offline Web Application

Denial of Service

- Test for anti-automation
- Test for account lockout
- Test for HTTP protocol DoS
- Test for SQL wildcard DoS

Specific Risky Functionality

File Uploads

- Test that acceptable file types are whitelisted and non-whitelisted types are rejected
- Test that file size limits, upload frequency and total file counts are defined and are enforced
- Test that file contents match the defined file type
- Test that all file uploads have anti-virus scanning in place
- Test upload of malicious files
- Test that unsafe filenames are sanitized
- Test that uploaded files are not directly accessible within the web root
- Test that uploaded files are not served on the same hostname/port
- Test that files and other media are integrated with the authentication and authorization schemas

Payments

- Test for known vulnerabilities and configuration issues on Web Server and Web Application
- Test for default or guessable password
- Test for Injection vulnerabilities
- Test for Buffer Overflows
- Test for Insecure Cryptographic Storage
- Test for Insufficient Transport Layer Protection
- Test for Improper Error Handling
- Test for all vulnerabilities with a CVSS v2 score > 4.0
- Test for Authentication and Authorization issues
- Test for CSRF

Error Handling

- Check for Error Codes
- Check for Stack Traces