Android Penetration Testing Checklist

Table of Contents-

SSL Pinning

- [ ] Verify if SSL Pinning is implemented.
- [ ] Test for bypassing possibilities using tools like Frida or Objection.
- [ ] Check if code manipulation (e.g., bypassing the certificate checks) is feasible.

Root Detection

- [ ] Ensure the app implements root detection.
- [ ] Test bypassing with Frida or Objection.
- [ ] Verify if root detection logic can be modified to access restricted data or functionality.

Emulator Detection

- [ ] Confirm emulator detection is implemented.
- [ ] Test bypassing using Frida or similar tools.

Sensitive Data in ADB Logcat Logs

- [ ] Check Logcat logs for sensitive information (e.g., passwords).
- [ ] Attempt bypassing Frida or Objection.
- [ ] Verify that unencrypted data does not appear in Logcat.

Sensitive Data Stored in Local Storage

- [ ] Examine Shared Preferences for sensitive information.
- [ ] Review temporary files for sensitive data storage.
- [ ] Ensure sensitive data is encrypted if stored locally (e.g., database).
- [ ] Check other files for any stored sensitive data.

Sensitive Data in Application Memory

- [ ] Inspect memory for sensitive data with tools like fridump.py.

Weak Signer Certificate

- [ ] Check for weak signing algorithms (e.g., SHA1withRSA).
- [ ] Identify vulnerabilities like Janus.
- [ ] Verify the app is signed with a production certificate, not a debug one.

Vulnerable Android Activities

- [ ] Ensure protected activities are not accessible directly via ADB.
- [ ] Confirm the `exported` attribute is set to `false` for non-public activities.
- [ ] Test if activities can be hijacked via ADB or other tools.
- [ ] Assess for denial of service (DoS) or app crash vulnerabilities in activities.

WebView Vulnerabilities

- [ ] Test WebView for Cross-Site Scripting (XSS) vulnerabilities.
- [ ] Check for Local File Inclusion (LFI) in WebView.
- [ ] Ensure JavaScript is only enabled when necessary and secured.

Intent Filters

- [ ] Inspect for intent spoofing and sniffing vulnerabilities.
- [ ] Confirm proper filtering of intent data to prevent Open Redirect-like issues.

Broadcast Receivers

- [ ] Check if receivers are set with `exported=true` and ensure additional permissions are applied if needed

Content Providers

- [ ] Validate content providers against SQL injection, path traversal, and unauthorized data access vulnerable

Source Code Obfuscation

- [ ] Ensure ProGuard or another obfuscation method is implemented.
- [ ] Confirm that sensitive parts of the code are fully obfuscated.

Sensitive Information/Auth-Keys Hardcoded

- [ ] Scan for hard coded sensitive data (e.g., API keys, tokens) using tools like MobSF.

Insecure Coding Practices

- [ ] Check for the use of insecure RNG functions for OTPs,etc.
- [ ] Inspect for unsafe functions or objects.
- [ ] Confirm cryptographic methods are secure (e.g., avoid MD5, Base64 for encryption).
- [ ] Review for other insecure coding weaknesses.

Insecure Deeplinks

- [ ] Test explicit deep links for potential exploitation.
- [ ] Review implicit deep links for unauthorized access.

Missing Integrity Checks

- [ ] Attempt to decompile, modify, recompile, and sign the app to see if it functions

  post-modification.

Insecure Android Permissions

- [ ] Review `AndroidManifest.xml` for cleartext traffic, debug mode, backup mode, and unnecessary perm
- [ ] Ensure `dataExtractionRules` are correctly defined.

Background Screen Caching

- [ ] Check if sensitive data is visible in screenshots when the app is

  minimized.

Insecure Firebase Database

- [ ] Append `.json` to the Firebase URL to test read permissions.
- [ ] Test replacing "firebaseio.com" with "appspot.com" and appending `.json` to check access permission

Android Lock/Biometric Authentication Bypass

- [ ] Test for bypassing lock or biometric authentication via runtime hooking or code modification.

Key Checks in Dynamic Analysis

- [ ] Perform API testing and fuzzing to verify access control and data exposure.
- [ ] Test for injection vulnerabilities and misconfigurations.

Additional Checks:

- [ ] Ensure the same cryptographic key is not reused across multiple contexts.
- [ ] Verify sensitive data isn't exposed through the UI or screenshots.
- [ ] Ensure keyboard caching is disabled where necessary.
- [ ] Prevent copy-pasting of sensitive data fields.
- [ ] Confirm sensitive data is masked during app switching.