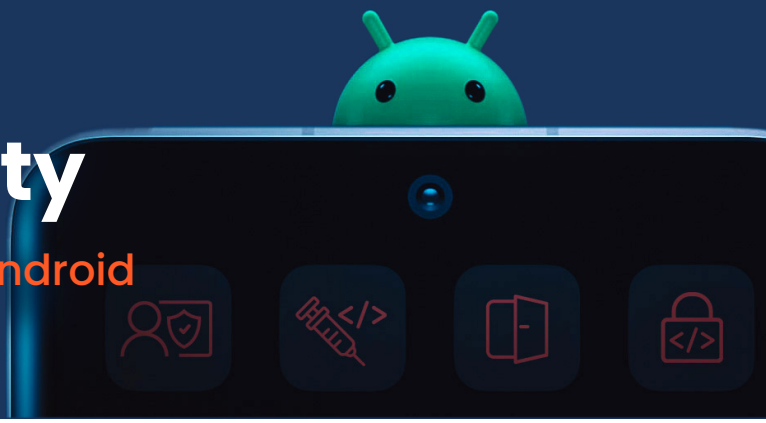
ZIMPERIUM

Mobile App Security

Critical Vulnerability Checklist for Android



In the rapidly evolving landscape of mobile app security, it's imperative for developers and security engineers to stay vigilant against potential vulnerabilities. This checklist is meticulously designed to guide you through some critical aspects of securing your Android mobile app. From safeguarding sensitive data to implementing robust encryption, each item serves as a crucial checkpoint to help ensure your application stands resilient against threats, maintains user trust, and complies with the highest security standards in the industry. Let's ensure your app's defenses are as robust and impenetrable as possible.

Outlined below are the essential steps developers should take to ensure the security of their mobile apps.

Debug and Development Configuration

- [ ] Ensure debug information is stripped from the release version of the app.
- [ ] Verify the 'android: debuggable' attribute is set to 'false' in the production manifest.

Application Shielding

- [ ] Confirm code obfuscation techniques have been implemented to protect against reverse engineering.
- [ ] Ensure that mechanisms are in place to detect and address attempts of code tampering.
- [ ] Implement integrity protection measures to make code and read-only data challenging to modify.

Cloud Storage and API Security

- [ ] Secure Amazon S3 buckets against unauthorized file and directory listing.
- [ ] Ensure FirebaseIO and Google Storage configurations do not allow world-viewable files.
- [ ] Implement secure storage solutions for API keys to prevent unauthorized access.
- [ ] Confirm that Azure Storage containers do not permit anonymous access.

Endpoint and Data Transfer Security

- [ ] Regularly audit and secure endpoints to prevent malware distribution risks.
- [ ] Confirm SSL communication uses secure methods and avoid using the `getInsecure` API.
- [ ] Establish a proper chain of trust validation for all endpoints.

Input and User Data Protection

- [ ] Replace any visible password input types with more secure input methods.
- [ ] Prevent saving device identifiers to external storage.

Cryptography and Key Management

- [ ] Secure cryptographic key management, especially private keys.
- [ ] Avoid exposing secret keys in the code and consider using a secure keystore.
- [ ] Determine appropriate sizes for encryption keys to mitigate risk.
- [ ] Avoid using encryption schemes with known vulnerabilities.
- [ ] Ensure cryptographic keys are safe even on compromised mobile devices.

Code and Dependency Security

- [ ] Ensure there's no inclusion of exploitable code, such as Metasploit code.
- [ ] Update the OpenSSL library to a non-vulnerable version.
- [ ] Remove any references to remote servers with known vulnerabilities, like FREAK.

Data Storage and Permissions

- [ ] Limit access to Content Providers and ensure they are not exported unless protected by signature-based permissions.
- [ ] Audit for undeclared permissions and ensure they align with the app's privacy policy.
- [ ] Control access to content providers, taking into account `android:grantUriPermission` settings.

SQL and Code Injection Protection

- [ ] Use parameterized queries or equivalent protections to prevent SQL injection vulnerabilities.

Ad Platforms and Third-party Services

- [ ] Review third-party ad platforms, such as Igexin, for privacy concerns and vulnerabilities.

Backdoor and Unintended Access Prevention

- [ ] Implement safeguards against potential backdoor threats and ensure only secure intent redirection.

File Handling and Integrity

- [ ] Set up permissions and safeguards to prevent unauthorized file writes or tampering with executable files.

Network Communication

- [ ] Enforce consistency in protocol handling, especially with the Apache HttpClient.
- [ ] Configure TrustManager to validate server certificates accurately.

Error Handling

- [ ] Handle SSL/TLS communication errors securely without bypassing them.

Malware Protection

- [ ] Review and remediate any malware findings within open-source and third-party components.

Privilege and Role Management

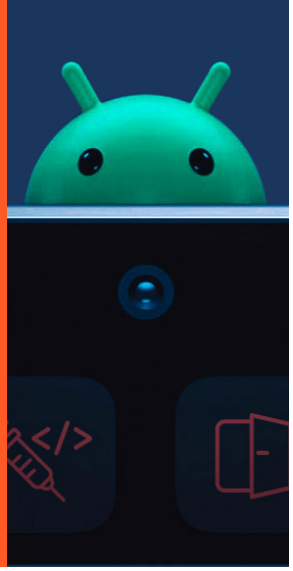
- [ ] Secure app resource values that are used as predicates for important privileges like admin access.

Software Bill of Material

- [ ] Review the software bill of material (SBOM) for any known issues and vulnerabilities within third-party code.

Running on Rooted/Emulated Device

- [ ] Implement measures to identify if the application is running on an emulator or rooted device to prevent potential tampering by malicious actors.



Instructions for Use

1. Go through each item on the checklist before releasing the app and regularly post-release.
2. Mark items as complete only when fully addressed and validated.
3. Use this checklist in conjunction with a comprehensive security audit for best results.
4. If any item on the checklist raises concerns, seek further review or remediation steps.

Next Steps

While this checklist aims to highlight key security issues, it's important to recognize that it is not exhaustive. Security is a multifaceted, ever-changing field, and continuous vigilance is required. Use this checklist as a starting point for your security review process and supplement it with ongoing education, keeping abreast of the latest security trends, threats, and best practices to protect your mobile application comprehensively.

Contact Our Security Experts for A Comprehensive Risk Assessment