

## **Android App Security Checklist**

A checklist with security considerations for designing, testing, and releasing secure Android apps. It is based on the OWASP Mobile Application Security Verification Standard and Mobile Security Testing Guide. Follow the links on each checklist item for detailed instructions and recommendations.

## **Data Storage**

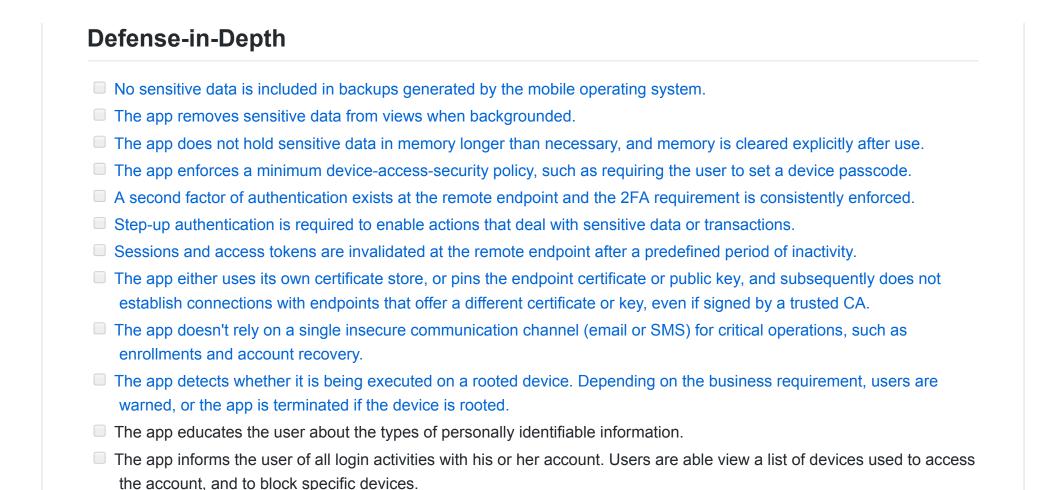
- The Keystore is used to store sensitive data, such as user credentials or cryptographic keys.
- No sensitive data is written to application logs.
- No sensitive data is shared with third parties unless it is a necessary part of the architecture.
- The keyboard cache is disabled on text inputs that process sensitive data.
- The clipboard is deactivated on text fields that may contain sensitive data.
- No sensitive data is exposed via IPC mechanisms.
- No sensitive data, such as passwords or pins, is exposed through the user interface.

## **Platform Interaction**

- The app only requests the minimum set of permissions necessary.
- All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.
- The app does not export sensitive functionality via custom URL schemes without proper protection.
- The app does not export sensitive functionality through IPC facilities without proper protection.

JavaScript	is disabled in WebViews unless explicitly required.
	are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). y dangerous handlers, such as file, tel and app-id, are disabled.
If native me package.	ethods of the app are exposed to a WebView, that WebView only renders JavaScript contained within the app
Object seri	ialization, if any, is implemented using safe serialization APIs.
Cryptogr	aphy
☐ The app do	oes not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.
☐ The app us	ses proven implementations of cryptographic primitives.
	ses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that industry best practices.
☐ The app do	oes not use cryptographic protocols or algorithms that are widely considered depreciated for security purposes
All random	values are generated using a sufficiently secure random number generator.
☐ The app do	oesn't re-use the same cryptographic key for multiple purposes.
Authentic	
	provides users with access to a remote service, an acceptable form of authentication such as performed at the remote endpoint.
A passwor	d policy exists and is enforced at the remote endpoint.
	e endpoint implements an exponential back-off, or temporarily locks the user account, when incorrect ation credentials are submitted an excessive number of times.
	session management is used, the remote endpoint uses randomly generated session identifiers to authenticate uests without sending the user's credentials.

<ul> <li>If stateless token-based authentication is used, the server provides a token signed using a secure algorithm.</li> <li>The remote endpoint terminates the existing stateful session or invalidates the stateless session token when the user logs out.</li> </ul>
■ Biometric authentication, if any, is not event-bound (i.e. using an API that simply returns "true" or "false"). Instead, it is based on unlocking the Keystore.
Network
■ Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.
The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.
■ The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.
Code Quality
Code Quality  The app is signed and provisioned with valid certificate.
■ The app is signed and provisioned with valid certificate.
<ul> <li>The app is signed and provisioned with valid certificate.</li> <li>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</li> </ul>
<ul> <li>The app is signed and provisioned with valid certificate.</li> <li>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</li> <li>Debugging symbols have been removed from native binaries.</li> </ul>
<ul> <li>The app is signed and provisioned with valid certificate.</li> <li>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</li> <li>Debugging symbols have been removed from native binaries.</li> <li>Debugging code has been removed, and the app does not log verbose errors or debugging messages.</li> </ul>
<ul> <li>The app is signed and provisioned with valid certificate.</li> <li>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</li> <li>Debugging symbols have been removed from native binaries.</li> <li>Debugging code has been removed, and the app does not log verbose errors or debugging messages.</li> <li>The app catches and handles possible exceptions.</li> </ul>



© 2018 GitHub, Inc. Terms Privacy Security Status Help



Contact GitHub API Training Shop Blog About