Features Business Explore Marketplace Pricing This repository Search Sign in or Sign up

b-mueller / **android_app_security_checklist**

Watch 25 Star 392 Fork 79

Code Issues 1 Pull requests 0 Projects 0 Insights

Dismiss

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Sign up

Android App Security Checklist

35 commits 1 branch 0 releases 2 contributors

Branch: **master** New pull request Find file Clone or download

b-mueller Update links Latest commit `4ab1016` on Jan 17

README.md	Update links	4 months ago

README.md

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.
- [ ] WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.
- [ ] If native methods of the app are exposed to a WebView, that WebView only renders JavaScript contained within the app package.
- [ ] Object serialization, if any, is implemented using safe serialization APIs.

Cryptography

- [ ] The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.
- [ ] The app uses proven implementations of cryptographic primitives.
- [ ] The app uses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that adhere to industry best practices.
- [ ] The app does 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 doesn't re-use the same cryptographic key for multiple purposes.

Authentication

- [ ] If the app provides users with access to a remote service, an acceptable form of authentication such as username/password authentication is performed at the remote endpoint.
- [ ] A password policy exists and is enforced at the remote endpoint.
- [ ] The remote endpoint implements an exponential back-off, or temporarily locks the user account, when incorrect authentication credentials are submitted an excessive number of times.
- [ ] If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.

- [ ] If stateless token-based authentication is used, the server provides a token signed using a secure algorithm.
- [ ] The remote endpoint terminates the existing stateful session or invalidates the stateless session token when the user logs out.
- [ ] 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

- [ ] The app is signed and provisioned with valid certificate.
- [ ] The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).
- [ ] Debugging symbols have been removed from native binaries.
- [ ] Debugging code has been removed, and the app does not log verbose errors or debugging messages.
- [ ] The app catches and handles possible exceptions.
- [ ] Error handling logic in security controls denies access by default.
- [ ] In unmanaged code, memory is allocated, freed and used securely.
- [ ] Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.

Defense-in-Depth

- [ ] No sensitive data is included in backups generated by the mobile operating system.
- [ ] The app removes sensitive data from views when backgrounded.
- [ ] The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.
- [ ] The app enforces a minimum device-access-security policy, such as requiring the user to set a device passcode.
- [ ] A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.
- [ ] Step-up authentication is required to enable actions that deal with sensitive data or transactions.
- [ ] Sessions and access tokens are invalidated at the remote endpoint after a predefined period of inactivity.
- [ ] The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.
- [ ] The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.
- [ ] The app detects whether it is being executed on a rooted device. Depending on the business requirement, users are warned, or the app is terminated if the device is rooted.
- [ ] The app educates the user about the types of personally identifiable information.
- [ ] The app informs the user of all login activities with his or her account. Users are able view a list of devices used to access the account, and to block specific devices.