OWASP Mobile Top 10

Extraneous Functionality

developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment.

Reverse Engineering

- Clearly understand the contents of a binarys string table
- Accurately perform cross-functional analysis

Derive a reasonably accurate recreation of the source code from the binary Although most apps are susceptible to reverse engineering, its important to examine the potential business impact of reverse engineering when considering whether or not to mitigate this risk

Code Tampering

all mobile code is vulnerable to code tampering. Mobile code runs within an environment that is not under the control of the organization producing the code. At the same time, there are plenty of different ways of altering the environment in which that code runs. These changes allow an adversary to tinker with the code and modify it at will.

Client Code Quality

This is distinct from Improper Platform Usage because it usually refers to the programming language itself (e.g., Java, Swift, Objective C, JavaScript). A buffer overflow in C or a DOM-based XSS in a Webview mobile app would be code quality issues.

The key characteristic of this risk is that it' s code executing on the mobile device and the code needs to be changed in a fairly localised way. Fixing most risks requires code changes, but in the code quality case the risk comes from using the wrong API, using an API insecurely, using insecure language constructs, or some other code-level issue

Insecure Authorization

- Presence of Insecure Direct Object Reference (IDOR) vulnerabilities
- Hidden Endpoints
- User Role or Permission Transmissions

Improper Platform Usage

- Violation of published guidelines
- Violation of convention or common practice
- Unintentional Misuse

Insecure Data Storage

- SQL databases
- Log files
- XML data stores ou manifest files
- Binary data stores
- Cookie stores
- SD card
- Cloud synced

Insecure Communication

- This risk includes all communications technologies that a mobile device might use: TCP/IP, WiFi, Bluetooth/Bluetooth-LE, NFC, audio, infrared, GSM, 3G, SMS, etc.
- The usual risks of insecure communication are around data integrity, data confidentiality, and origin integrity.

Insecure Authentication

- If the mobile app is able to anonymously execute a backend API service request without providing an access token, this application suffers from insecure authentication.
- If the mobile app stores any passwords or shared secrets locally on the device, it most likely suffers from insecure authentication.
- If the mobile app uses a weak password policy to simplify entering a password, it suffers from insecure authentication.
- If the mobile app uses a feature like TouchID, it suffers from insecure authentication

Insufficient Cryptography

- The mobile app may use a process behind the encryption / decryption that is fundamentally flawed and can be exploited by the adversary to decrypt sensitive data.
- The mobile app may implement or leverage an encryption / decryption algorithm that is weak in nature and can be directly decrypted by the adversary.