MANAGEMENT MANIA

Akshit Rana

Android App Development | Best Practices to Enforce Secure Communications

Android app or software development is a process of creating applications for devices running on Android operating systems. The developers can write Android applications in different languages such as C++, Java, and Kotlin while exploiting the Android SDK (software development kit).

When developing an application, there are multiple facets of the process including designing, security, integration of features, UX, and multiple others. Out of all these facets, the one that I am going to discuss here is the security of Android applications while communicating internally or with end-users.

***Today's world is of technological advancements. If there are splurging digital solutions, there is an equally increasing demand for security. Hackers, nowadays, are smart enough to break the system and steal valuable information regarding a brand or any of its consumers. Therefore, it becomes important for every business to prioritize security when planning to _hire Android developer*. After all, it is the responsibility of a developer to inject security elements into applications, starting from scratch._**

In this post, I shall discuss best security practices to enforce secure communication of Android applications. Let's begin!

Data Security

While developing Android applications there is a need to utilize Android's built-in security library. This library facilitates easy and effective implementation of best security practices in context with data writing and reading, along with creating and verifying keys. Under this library majorly two security levels are covered:

Great encryption and good app performance

The extreme level of security

To learn more about the data security capabilities of Android SDK, click here. Also, you need to look for suitable Android app development tools and methods to ensure complete data security.

HTTPS and SSL

To ensure encrypted communications between servers and clients, SSL (secure sockets layers) or TLS (transport layer security) is the most popular approach. There are chances that an Android application might be using SSL improperly. To eradicate the risk of security vulnerabilities in such scenarios, you need to watch out for some common loopholes in your Android app structure as mentioned below:

- ***Unknown certificate authority.***
- ***Self-signed certificate.***
- ***Missing intermediate certificate authority.***

To address such issues, you can implement SSL pinning, Handling compromise, Handling private key leaks, etc. For details, you can refer to the official documentation of Android developers.

Protecting against SSL exploits:

To provide secure network communications, the Android platform hugely relies on a security provider. However, there are fair chances of vulnerabilities in your chosen security provider, from time to time. To deal with these, Google Play Services facilitates you with automatic security provider updation of devices, against SSL exploits.

Additionally, you can use 'ProviderInstaller' to patch the device's security provider. Call the class's "installIfNeeded()" "or installIfNeededAsync()" method to verify that the device's security provider is up-to-date. If not updated, do it instantly.

Miscellaneous App Security Practices

Apart from the above-mentioned practices, there are some other tips and tricks too that can facilitate secure network communications of Android applications. This section contains multiple best security practices that will leave a positive and significant impact on your Android app's security. Let's begin!

- ***Safeguarding external communications (between your app and other apps or other websites). Doing this will allow you to***

ensure the app's stability and to protect data while sending and receiving.

Utilize content providers (non-exported) and implicit intents.

Incorporate signature-based permissions.

Prohibit access to your Android app's content providers.

Request user credentials for accessing sensitive information.

Opt for SSL traffic.

Build your own trust manager.

Carefully use WebView objects.

Final Words

Making your Android application is imperative as it helps you to preserve the user trust and integrity of the device. This post contains major security practices that help you impose proven security practices in your app. Apart from that, you can always look for custom security solutions with the help of experts.