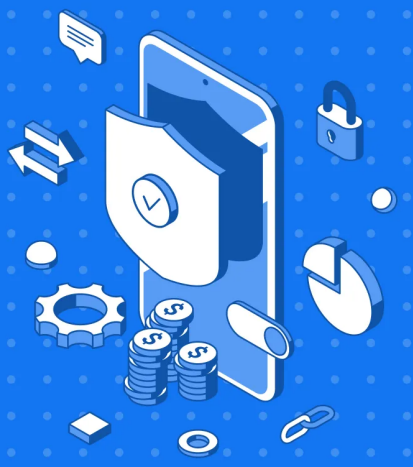
magora

MOBILE APPLICATION SECURITY:

HOW TO PROTECT ANDROID APPS FROM REVERSE ENGINEERING



Safety is an essential part of any product's quality. Unfortunately, in the high-tech world an application can be an easy target for hackers.

The first thing we worry about is personal data safety. However, someone may also wish to hack it (device, program, software) in order to understand the principle of its operation and adopt ideas, or even completely reproduce the application. This practice is called reverse engineering and is used in many areas, including electronics and even the military industry.

With respect to Android, reverse engineering is the process of extracting source code and resources from an APK (Android Package Kit) file. The APK of the target application can be removed from the phone by using ADB (Android Debug Bridge) or downloading from the Google Play Market.

Problem

Decompiling an APK file is not a difficult task. Through the decompilation of APK and the conversion of dex files to jar files and then to Java source code, hackers can acquire the source code of your app.

The fact is, it is almost impossible to completely protect a mobile app from reverse engineering and ensure its complete safety.

Nevertheless, here at Magora we strive to complicate the process of hacking as much as possible. In this article, we share our practical experience in protecting Android apps from reverse engineering.

Solutions

magora

+44 20 7183 5820

info@magora.co.uk
sales@magora.co.uk

1. Using ProGuard

To provide mobile application security from reverse engineering we can use the ProGuard tool, designed to reduce, optimise and obfuscate the code. It works as follows:

- analysis and optimisation of bytecode methods;
- detection and removal of unused classes, fields, methods and attributes;
- renaming of other classes, methods and fields using short, meaningless names.

These actions reduce the code base, making it more efficient and at the same time difficult to decrypt. At the final stage of the preliminary verification, information is added to the classes required for Java Micro Edition, Java 6, etc.

It is worth noting that obfuscation can be canceled with a deobfuscator. In this case APK De-Guard is one of the most accurate tools, as it uses machine learning to get the best results.

2. Moving Important Parts of the Code to the Server

Speaking of other effective methods, we also move the most important parts of the service from the mobile application to a web service hidden on the server side.

Imagine that you have a unique algorithm. Obviously you don’t want it to be stolen. Therefore, you can move it by making it process the data on the remote server and use the application to provide it with this data.

3. Writing Important Code in C / C ++

We sometimes use NDK (Native Development Kit) to write algorithms originally in .so files, which are much less likely to be decompiled than APK. In addition, we use SSL (Secure Sockets Layer) protocol when communicating between the device and the server.

Although it can be parsed into assembly code, the process of reversing the engineering of a large library from the assembly is quite laborious. In terms of security, compared with C / C ++, Java is easier to decompile.

4. Do not store values in an unprocessed format

Next, when storing the values on the device, we do not use the raw format. If we need to maintain the user's balance (the amount of the application currency) or other values, we usually store the values encoded (for example, store them in the form of an algorithm).

5. User Credentials

The next step towards your mobile application’s security relates to user account data.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

- First, we try to minimise the frequency of requests for user credentials in the app. This will help make phishing attacks more visible and reduce the likelihood of their success. In this case, we recommend using the authorisation token (and regularly updating it).
- Secondly, the name and password should not be stored on the device (where possible). We always advise performing the authorisation process with a username and password and using the authorisation token.

If you need to allow users to store their credentials to automate future authorisation in the app, we use a Credential object that contains information about the user's registration.Conclusion

Although it is almost impossible to guarantee 100% security of the application from reverse engineering and other threats, here at Magora, we do everything possible to protect your data. Here is the general list of recommendations to provide the highest level of security for your mobile app:

- Inform programmers about security issues.
- Factor security into the architecture.
- Perform an audit of the code.
- Apply compiler-related security settings.
- Control the distribution of the app on the Internet.
- Update regularly.

If you want to know more about the security measures we take or if you need a highly protected mobile app for business, our team is always ready to answer all your questions.

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk