Final Project
Network Security
Prof. Diego F. Aranha

1 Introduction

The objective of this project is to perform and document the results of a preliminary security analysis of a mobile application. The application should have a sensitive role impacting user privacy to motivate the security analysis. By sensitive role, it means that the application should handle user data of some sensitive nature (financial information of any kind, health records, location data, passwords and other authentication information, etc). Candidate applications can be of multiple types, for example:

- Financial (for banking and payments like MobilePay and local banks)
- Healthcare (like COVID apps for decentralized contact tracing)
- e-Government (nemID, eBoks and other digital government efforts)
- Transport (Midttrafik and Flixbus apps where you can buy tickets)
- Social Networks (TikTok, Tinder and similar, preferably a local one)

Try to choose an app for which development practices are likely not state-of-the-art, so you have a chance to find interesting issues to discuss. You can download APK files containing the application for analysis through one of the numerous repositories online. APKMirror is frequently recommended as the best free website for the task[1].

2 Method

Using the techniques and concepts you have learned in class, you should analyze the application in four different aspects:

1. *Software security*: decompile the application, look for keywords (such as cryptographic algorithms) or scan the source code using static analyzers.

[1] `https://www.apkmirror.com/`

2. *Network security*: document the TLS server configuration and attempt to MITM a connection to capture sensitive traffic. Observe the exchanged traffic for sensitive data.

3. *Authentication*: document what authentication mechanisms are used, and how secure they are.

4. *Privacy*: observe relevant privacy characteristics (trackers and integration with social networks).

The analysis should be performed under a relevant threat model and specification of related security properties. Make sure to formalize those in your report.

3 Material

During the course we discussed a multitude of techniques to perform the analysis above, summarized below for reference:

1. *Software security*: Android applications can be easily decompiled through APKTool[2], dex2jar[3], JG-GUI[4] or JDAX[5]. Notice that JAVA code is surprisingly easy to understand after decompilation, but obfuscation is also commonly found in production applications. For this reason, you might get slightly different results from these tools. In terms of static analysis, MobSF[6] produces quite detailed reports. Document what you find.

2. *Network security*: After capturing the server hostnames from the source code or using Wireshark, the TLS server configuration can be quickly analyzed through the Qualys SSL Labs[7]. Man-in-the-middle (MITM) attacks can be attempted with mitmproxy[8], sslsplit[9] (both with or without a self-signed root certificate) or Burp Suite[10]. Traffic can be redirected to the attacker's machine by simply configuring the malicious IP address as the gateway or by running the application inside the Android emulator.

3. *Authentication*: Authentication security can be analyzed by inspecting captured traffic in case the MITM attacks are successful, or following authentication workflows in the application (password recovery/renewal, availability of multi-factor authentication).

[2]https://ibotpeaches.github.io/Apktool/
[3]https://sourceforge.net/projects/dex2jar/
[4]https://java-decompiler.github.io/
[5]https://github.com/skylot/jadx
[6]https://github.com/MobSF/Mobile-Security-Framework-MobSF
[7]https://www.ssllabs.com/
[8]https://mitmproxy.org/
[9]https://www.roe.ch/SSLsplit
[10]https://portswigger.net/

4. *Privacy*: Use the theoretical tools we used in class for a qualitative assessment. Privacy-invasive behavior can also be analyzed by observing data collection practices, tracker traffic, requested permissions and deep integration with social networks. GDPR compliance and draconic Terms of Use are also concerns.

Observation: Decompilation for security analysis is a gray area in Europe due to the 2009/24/EC Directive, but our purpose is legitimate and compatible with broad interpretations of the law[11]. Let me know if you want to discuss specific concerns about this issue.

4 Evaluation

Write a report containing your observations. Use screenshots to document the most interesting bits you find. The submission deadline is **19/12/2021**. The instructor will be available for support through e-mail or in-person meetings during the Wednesday 8-10 time slot previously devoted to lab exercises.

The project can be done individually or in groups of at most 3 members. For the purposes of grading, the report should include a description about the distribution of work among the group members, highlighting the contributions of each member. The instructor and censors reserve the right to discuss individual participation during the oral examination in January.

The expected length of the report is 7 pages for individual work, and around 3 additional pages per group member (in double-column format). Groups are expected to deliver broader and deeper analysis, for example attempting multiple approaches for analysis in case the first one fails. Submissions must be performed through Brightspace. Sample reports from 2019 can be found attached to the corresponding assignment page.

[11] `https://www.technologylawdispatch.com/2021/10/in-the-courts/ecj-top-system-ruling-grants-right-to-correct-software-errors/`