**UNIVERSITY OF LUXEMBOURG**

**FACULTY OF SCIENCE, TECHNOLOGY AND COMMUNICATION**

**Self-defense capabilities for Android applications against** **repackaging**

Mentor

**Aleksandr PILGUN**

Student  
**Davit POGOSIAN**

Student id

**017104145E**

# 1. Introduction

Android is the most popular mobile operating system. In 2017, 66,74% of all smartphones were using Android as an operating system and over 330000 applications are available in Google Play for this platform. One of the top priorities of Android is openness, as open that the users can install applications from third-party markets. That is why the number of Android applications is growing rapidly. As an effect of openness, Android apps can be easily decompiled due to their structural characteristics, which mean that apps modified in this way can be repackaged.

Hackers can add or modify logics of the original apps by using some well-known tools and then release the modified app to a third-party market as new version. Researchers found out that 86% of malwares are repackaged malware, which demonstrates the popularity and severity of repackaged malware. Malwares are not the only problem based on repackaging. Hackers can modify original applications and stole information such as personal data, bank accounts and so on. This is why (That is why?) repackaging is a big problem in the Android ecosystem.

This paper is organized as follows. Section 2 takes a brief look at related work. Section 3

describes the results of a vulnerability analysis performed on actual banking apps. Section 4

describes the attack results of the forged app, which was actually published on the Android

Market. Section 5 attempts to find countermeasures for the repackaging vulnerability. Finally,

the conclusions are presented in Sect. 6.

# 2. Related work

## 2.1 Java

## 2.2 APK



# 3. Repackaging

For this project I made differet tests to repackage my own application (Hello Romania) APK and Mobiliteit.lu APK.

Mobiliteit.lu app is a very popular application being used for information regarding bus/train schedules in the Grand Duchy of Luxembourg.

To repackage Mobiliteit.lu application I use APKTool and for my own application I used dex2Jar and JD-GUI, tools which I will present below.

## 3.1 APKTool

To repackage an application using APKTool you only need the APKTool itself, which is free and can be downloaded without any restrictions and the target application APK.

The Mobiliteit.lu application has been downloaded from the Google Play Store and using ES File Explorer I created a backup of the application which then created an APK file on the SD card.

Using the command **java -jar apktool\_2.3.0.jar d -f -o *path\_for\_decoded\_files* *path\_for\_APK*** the APK was decoded and in the folder containg the decoded files, among other files, we have:

* the *AndroidManifest.xml* which is unencrypted and readable using a notepad similar application,
* in the smali folder, we can find the application’s code in smali language,
* in the res folder, we can find all resources, which are decoded.

To test repacking on this application, I decided to change its colors, which was made by modifying the *colors.xml* file contained under **res\values** folder, using the value **#800000** for most of the colors.

The next step was to use the command **java -jar apktool\_2.3.0.jar b *path\_for\_decoded\_files*** for repackiging the application, which created the modified APK file named *Mobiliteit.apk*.

For signing the new APK file, I used ZipSigner application.

To test the success of the repackiging, I installed the modified APK on my device and below you can see the results: the first capture is of the original application and the second on if from the modified and repackaged application:

 