uc3m | Universidad Carlos III de Madrid

Grupo de investigación:
Computer Security Lab

Miguel Ángel Díaz Bautista
Roberto Blanco Ancos

Module II Assignment: Reverse Engineering and Repackaging

Mobile Devices Security

Table of Contents

1 Introduction

The objective of this module is to apply some techniques used by developers who want to analyze their app’s security and/or by malicious developers who intend to modify that app. Someone using the techniques covered in this module usually pursues one of the following goals (which may become part of an illegal activity):

1. To add malicious behavior
2. To bypass an intended functionality

The goals established for this assignment are as follows:

- Understanding how a particular apk is decompiled and repackaged
- The general principle of obfuscation
- To have a general view of how we can obfuscate our code using the android SDK - ProGuard tool, which shrinks, optimizes, and obfuscates the code by removing unused code and renaming classes, fields, and methods with semantically obscure names.
- Usage of smali and baksmali features to unpack and repack an apk
- Traffic analysis of the data transmitted by an app, in order to find indicators of compromise
- Being able to ascertain the need for secure storage and transmission of sensitive data

Reverse engineering might be an illegal activity, thus it should only be performed on our own applications. The following tools will be of use in this module:

- Dex2jar: http://code.google.com/p/dex2jar/
- JD-GUI: http://java.decompiler.free.fr/?q=jdgui
- ApkTool: https://ibotpeaches.github.io/Apktool/
- ProGuard documentation: www.proguard.sourceforge.net
- 010 Editor: http://www.sweetscape.com/
- Notepad ++: http://notepad-plus-plus.org/
- Wireshark: https://www.wireshark.org/

After completing this practical assignment, you must have gained general skills for:

- Partial retrieval of source code from an apk
- Obfuscating source code of an app

- Repackaging of a particular apk, which would include some modifications that would make it behave differently from the original one
- Monitoring and analysis of data transmitted between apps and other devices on a network

2 Modification of the "CredHub" app

Your first task is to modify the *CredHub* app designed in module 1. The modifications will be as follows:

- The first activity, after the title screen, will be a login screen asking for a username and password (see Figure 2.1).
- If the authentication process succeeds, the user will be taken to the main activity.
- Those tasks requiring access to the remote repository will be modified in such a way that HTTP queries will contain an "`Authorization: Basic`" header that includes the username and password entered during the login phase.

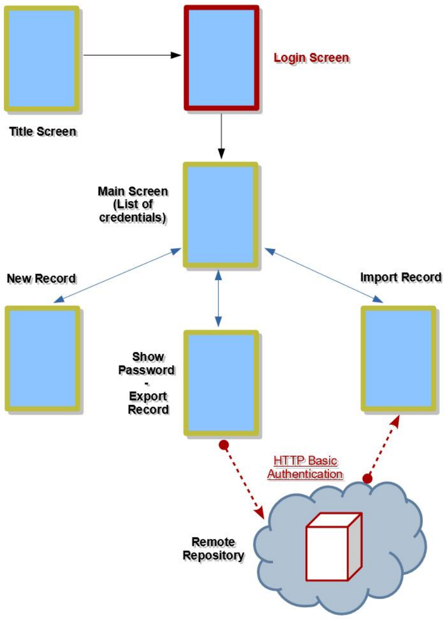
Login Screen
Title Screen
Main Screen (List of credentials)
New Record
Import Record
Show Password - Export Record
HTTP Basic Authentication
Remote Repository

Figure 2.1. Updated workflow of the *CredHub* app.

Login credentials (username and password) will have to be previously stored in a XML resources file of the app. The password cannot be stored in plaintext, but by means of a hash function, similarly to the process explained on the exercises for this module (app *uc3m_banca.apk*).

We must keep in mind that, for the web server to be able to handle these new queries that include authentication headers, the server´s executable file must be launched using the following input arguments:

```
"%JAVA_HOME%\bin\java" -jar SDM_WebRepo.jar http+auth
```

3 Reverse engineering and repackaging

This part of the assignment will consist of:

a. Obfuscating the developed app.
b. Decompile your apk and modify it using smali code, so that you are able to login with a different password from the original one.
c. Repackage the application with the changes performed on previous steps.
d. Re-sign the modified application and install it in the emulator.
e. Run the application in order to verify that the credentials have been changed successfully. Execute some of the functionalities requiring access to the remote repository, and observe whether the app behaves as expected or not.
f. Launch a version of the app created before the repackaging process (meaning, the apk resulting from the obfuscation performed on step *"a"*) on the emulator, having initiated the emulator itself through the following command:

```
emulator -tcpdump webtraffic.cap -avd <emulator_name>
```

After this, close the emulator and analyze the HTTP traffic generated inside the file *webtraffic.cap* using Wireshark: describe all the relevant information found during this analysis.

4 Module II - Delivery

This practice will be done by groups of two (same groups as those of Module 1's assignment).

Each group will deliver through AulaGlobal:

- A report in pdf format describing the work made for section 3 of this Module. Organization and presentation of the report is important: a cover page, a table of contents and conclusions are mandatory.
- The Android Studio project along with the modified *CredHub* app (without obfuscating). It shall include the login screen activity as well as the HTTP basic authentication header.
- The .apk file of the *CredHub* app recompiled and signed with all modifications applied.
- In a separate text file, the valid username and password.

Failure to follow instructions will negatively influence your mark on this assignment.