Exercises in Secure Mobile Systems

FH Hagenberg, WS 2022
FH-Prof. DI Dr. Erik Sonnleitner

Exercise 6: Reverse Engineering

Exercise 6.1: Understanding SMALI 1

(1) Reverse engineer the following Dalvik bytecode snippet into high-level Java code, and explain what the code does.

```
.method private mymethod([II)V
  aget         v0, v3, v4
  add-int/lit8 v1, v4, 0x1
  aget         v1, v3, v1
  aput         v1, v3, v4
  add-int/lit8 v1, v4, 0x1
  aput         v0, v3, v1
  return-void
.end method
```

- Notes:
  - First argument passed to method is **v3**, and of type **[I**
  - Second argument passed to method is **v4**, and of type **I**

Exercise 6.2: Understanding SMALI 2

(2) Reverse engineer the following Dalvik bytecode snippet into high-level Java code, and explain what the code does. Note: The Java code does not need to be a compilable project, but must be syntactically valid Java.

```
.method private mymethod2(Ljava/io/InputStream;)I
.catch Ljava/io/IOException; {:try_start_0 .. :try_end_0} :handler_0
  :try_start_0
  invoke-virtual {v2},Ljava/io/InputStream/read
  move-result v0
  :try_end_0
  return v0
  :handler_0
  move-exception v0
  const/4 v0,15
  goto :try_end_0
.end method
```

Exercise 6.3: Reverse engineering in real-life

(3) Choose any Android application available in the Google Play store (e.g. via **`apkpure.com`**), download and disassemble it. Analyze (and possibly modify) the SMALI bytecode in order to achieve a particular goal. If your goal requires code modification, re-package the APK and run it on an Android system. Interesting goals include, but are not limited to:

- Removing ad banners
- Transforming a "free" to a "paid" application (for research purposes only)
- Bypass authentication,
- Analyze code in order to understand authentication, credential storage, certificate checking, ...
- ...or anything else interesting really

Submission: Describe your goal and the exact process what you did, what you found, what problems you faced, how you got your findings, etc.

Note: Sometimes, APKs are *obfuscated*. If you disassemble code which looks weired (e.g. no human-readable method or class names), better switch to another app – reverse engineering is still possible, but much harder to achieve.

Exercise 6.4: CTF, round 2

- **Bonuses:** For all additional levels solved, you'll get 1% exam bonus for every **5 additional stars**. This works up to +6%. A complete walkthough write-up for every solved level is required, in addition to answering specific questions in person about certain details.