Handout: Easily Instrumenting Android Applications for Security Purposes

Eric Bodden (eric.bodden@cased.de)
Steven Arzt (steven.arzt@cased.de)
Siegfried Rasthofer (siegfried.rasthofer@cased.de)

TECHNISCHE UNIVERSITÄT DARMSTADT

EC SPRIDE
EUROPEAN CENTER FOR SECURITY AND PRIVACY BY DESIGN

1. Installing VirtualBox

Go to: https://www.virtualbox.org/wiki/Downloads
Download latest VirtualBox for your system
Download latest VirtualBox Extension Pack

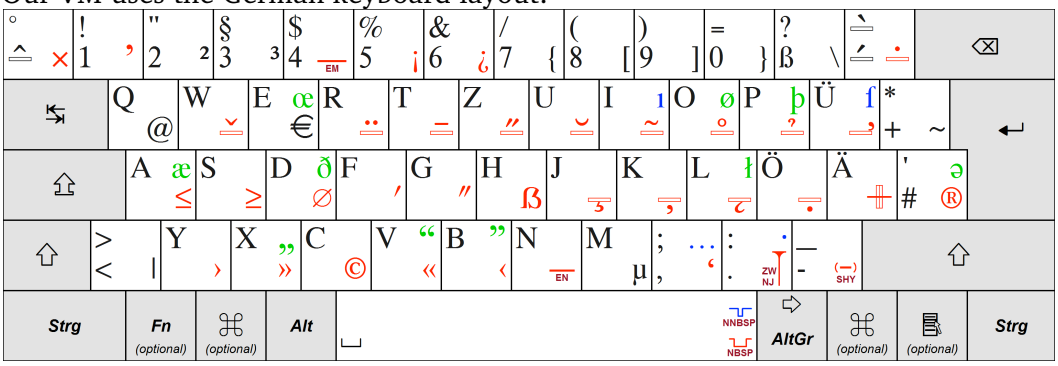
- Install VirtualBox 4.2.16 or later
- Install VirtualBox 4.2.16 or later Extension Pack

2. Getting Started with the VM

- Login:
  - User: rv2013
  - Password: rv2013

- Soot and abc path: /opt/soot
- Android SDK path: /opt/android-sdk-linux
- RV sample app path: ~/RV2013Examples/exampleApp

- Our VM uses the German keyboard layout:



3. Lab-Sessions:

Lab 1: Exploring and installing SMS Messenger app

- Open the RV2013 app in Eclipse
  - It should already be in your workspace
- Install it on the emulator
- Play around with it and look for Logcat outputs in Eclips

- Tip: Use Eclipse to install app or use “adb install RV2013.apk”
- Tip: If you need to remove the app: “adb uninstall de.ecspride”

Lab 2: Instrument an App using AspectJ

- Create an aspect that only allows 3 SMS messages per premium number, but an unlimited number of messages to normal numbers.

- Tip: Combine the aspects for the two policies.
- Tip: The app files are located under ~/RV2013Examples/exampleApp
- Tip: The aspect files are located under ~/RV2013Examples/aspectsAndTMs
- Copy and modify both the .sh and .aj file
- When invoking the .sh script the signature process will ask for a password. Just use rv2013

Lab 3: Instrument an App using Tracematches

- Change the tracematch such that it prevents SMS spam instead of just reporting it.

- Tip: Use an “around” advice. You don’t need to call “proceed” since your code is only called in the alert state.
- Tip: The app files are located under ~/RV2013Examples/exampleApp
- Tip: The tracematch files are located under ~/RV2013Examples/aspectsAndTMs
- Copy and modify both the .sh and .aj file

Lab 4: Analyze a Jimple Method

- Analyze the “reverseMe” method in the “RV2013” class and find out what it does.

- Tip: The original APK file is located under ~/RV2013Examples/RV2013.apk
- Tip: Look at the Jimple files generated by running soot with the output format set to “jimple”.

Lab 5: Insert a Premium-Rate SMS Check

- Before every call to sendTextMessage, check whether the phone number is a 0900 number. In case of a constant number just remove the statement otherwise skip the call. If it is not a 0900 number, proceed as normal.

- Tip: The following Jimple code snippets may be useful:

```
$z0 = virtualinvoke $r3.<java.lang.String: boolean startsWith(java.lang.String)>("0900“)
if $z0 == 0
goto nop

  virtualinvoke $r6.<android.telephony.SmsManager: void sendTextMessage(...)>

nop
```