Reversing & Protecting Android applications

#OWASPSpain8

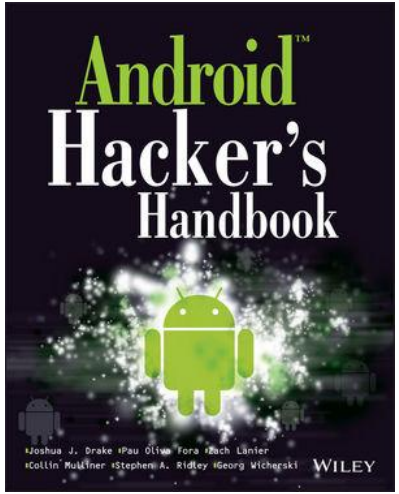
2014-06-13

Pau Oliva Fora

Sr. Mobile Security Engineer
viaForensics
@pof

$ whoami

- Pau Oliva Fora, aka @pof
  - Mobile security engineer with viaForensics
  - Linux guy, R+D background
  - Smartphone research since 2004
  - Android research since 2008
  - Co-author of Android Hacker's Handbook



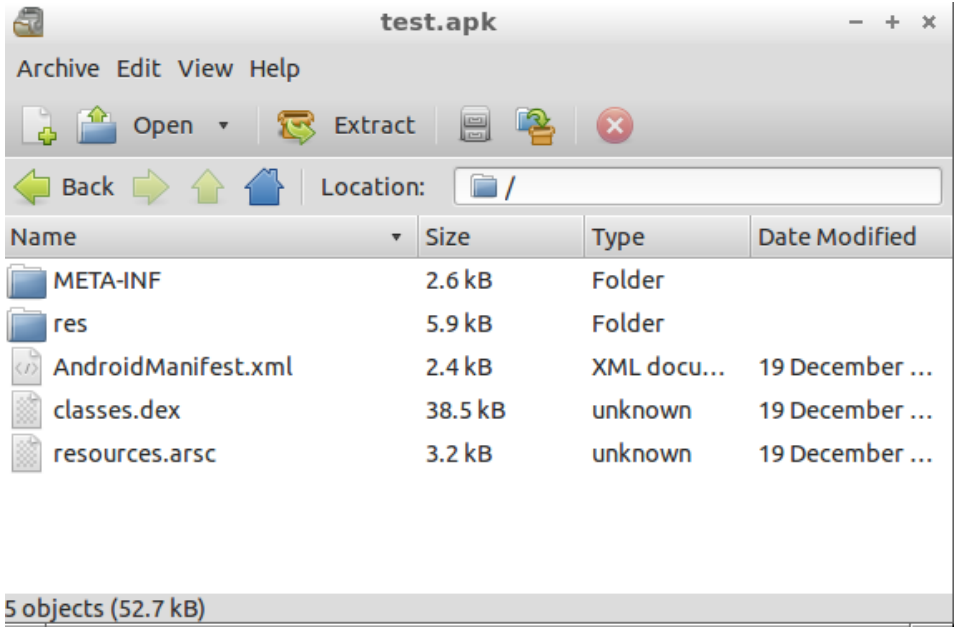
Agenda

- Reversing Android Apps
  - Anatomy of an Android app
  - Obtaining our target apps
  - Getting our hands dirty
  - Demo using Santoku Linux
- Protecting Android Apps
  - Common app vulnerabilities and FAILs
  - Building in-app security

Anatomy of an Android app

Anatomy of an Android app

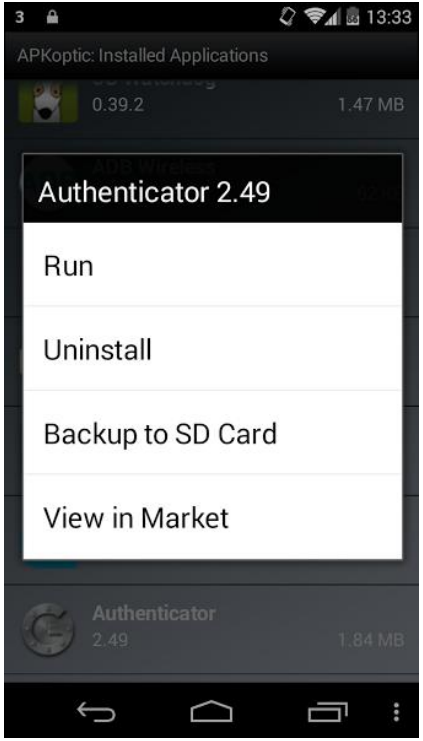
- Simple ZIP file, renamed to "APK" extension
- App resources
- Signature
- Manifest (binary XML)
- Bytecode (DEX)



Obtaining our target apps

Getting the APK from the phone

- Backup to SD Card:
  - APKOptic
  - Astro file manager
  - etc…



Getting the APK from the phone

- Using ADB (Android Debug Bridge):
  - adb shell pm list packages
  - adb pull /data/app/package-name-1.apk

```
santoku@santoku-VirtualBox:~$ adb shell pm list packages |egrep -v "(google|andr
oid)"
package:com.tf.thinkdroid.sg
package:es.vodafone.mobile.mivodafone
package:com.anydo
package:org.eslack.rootadb
package:com.saurik.substrate
package:com.viaforensics.cydiadynamicanalyzer
package:com.simyo
package:eu.chainfire.supersu
santoku@santoku-VirtualBox:~$ adb pull /data/app/com.simyo-1.apk
626 KB/s (1620854 bytes in 2.527s)
santoku@santoku-VirtualBox:~$
```

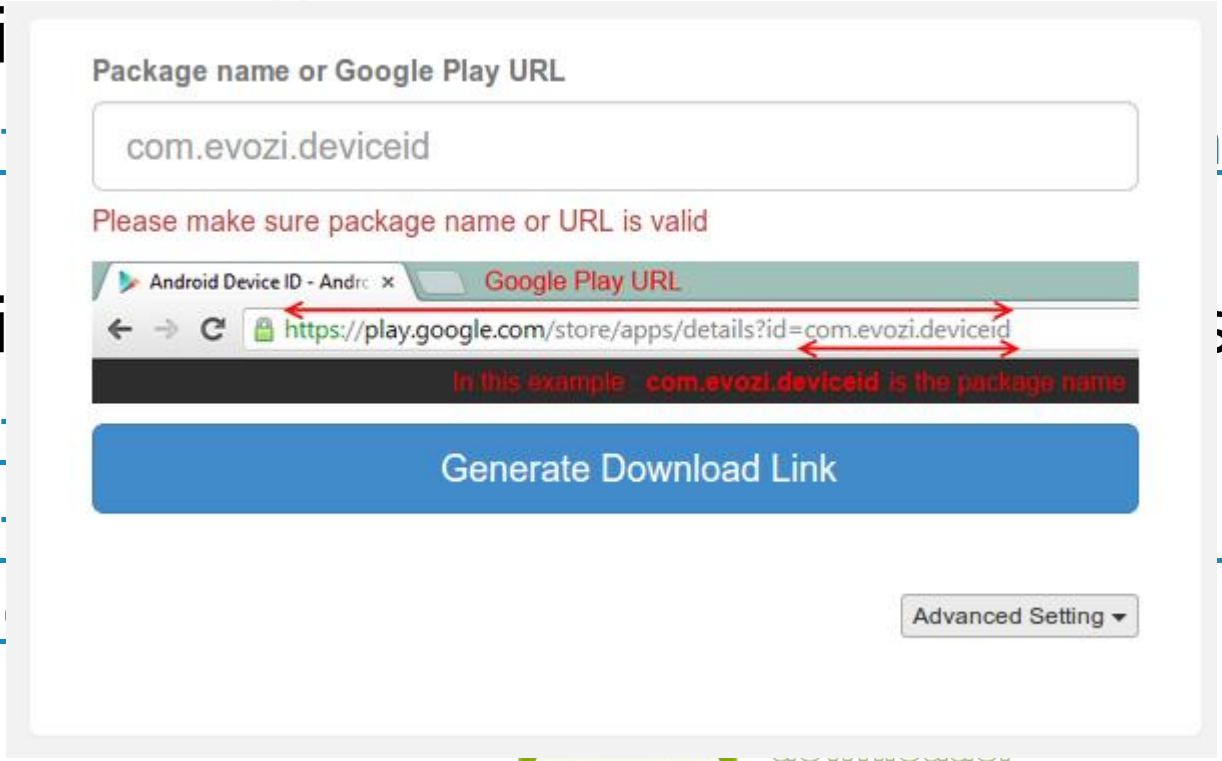
Downloading the APK from Google Play

- Using unofficial Google Play API:
  - https://github.com/egirault/googleplay-api
- Using a web service or browser extension:
  - http://apps.evozi.com/apk-downloader/
  - http://apify.ifconfig.com/static/clients/apk-downloader/

APK downloader

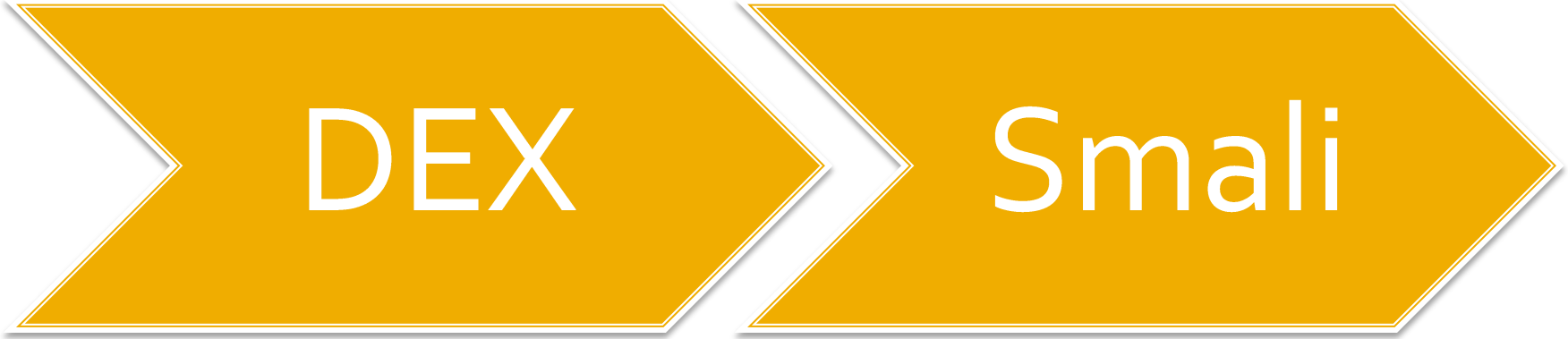
Downloading the APK from Google Play

- Usi
  - h pi
- Usi sion:
  - h
  - h k-
    d



Getting our hands dirty: reversing the target application

Disassembling



Apktool

- **apktool** - https://code.google.com/p/android-apktool/
  - Multi platform, Apache 2.0 license
  - Decode resources to original form (and rebuild after modification)
  - Transforms binary Dalvik bytecode (classes.dex) into Smali source

```
santoku@santoku-VirtualBox:/tmp/apk$ apktool d test.apk
I: Baksmaling...
I: Loading resource table...
I: Loaded.
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/santoku/apktool/framework/1.apk
I: Loaded.
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Done.
I: Copying assets and libs...
santoku@santoku-VirtualBox:/tmp/apk$ ls -l test/
total 16
-rw-rw-r-- 1 santoku santoku 1156 Jan  3 16:05 AndroidManifest.xml
-rw-rw-r-- 1 santoku santoku  262 Jan  3 16:05 apktool.yml
drwxrwxr-x 5 santoku santoku 4096 Jan  3 16:05 res
drwxrwxr-x 3 santoku santoku 4096 Jan  3 16:05 smali
santoku@santoku-VirtualBox:/tmp/apk$
```

Smali

santoku@santoku-VirtualBox: /tm.../smali/com/viaforensics/android

File Edit Tabs Help

```
    .prologue
    .line 55
    invoke-virtual {p0}, Ljava/lang/Object;->getClass()Ljava/lang/Class;

    move-result-object v0

    invoke-virtual {v0}, Ljava/lang/Class;->getName()Ljava/lang/String;

    move-result-object v0

    const-string v1, "Error message: "

    invoke-static {v0, v1, p1}, Landroid/util/Log;->e(Ljava/lang/String;Ljava/la
ng/String;Ljava/lang/Throwable;)I

    .line 56
    invoke-virtual {p1}, Ljava/lang/Exception;->getMessage()Ljava/lang/String;

    move-result-object v0

    invoke-virtual {p0, v0}, Lcom/viaforensics/android/ExtractAllData;->showErro
rOccuredToast(Ljava/lang/String;)V

                                                          25,5          22%
```

Decompiling – Java Decompiler

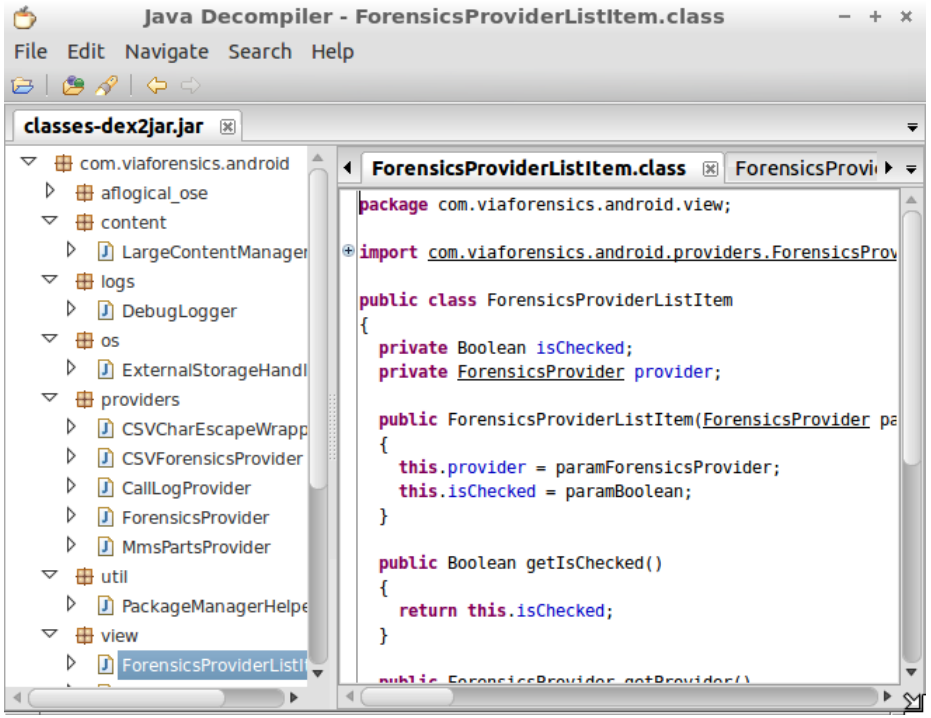
Dex2Jar

- **dex2jar** - https://code.google.com/p/dex2jar/
  - Multi platform, Apache 2.0 license
  - Converts Dalvik bytecode (DEX) to java bytecode (JAR)
  - Allows to use any existing Java decompiler with the resulting JAR file

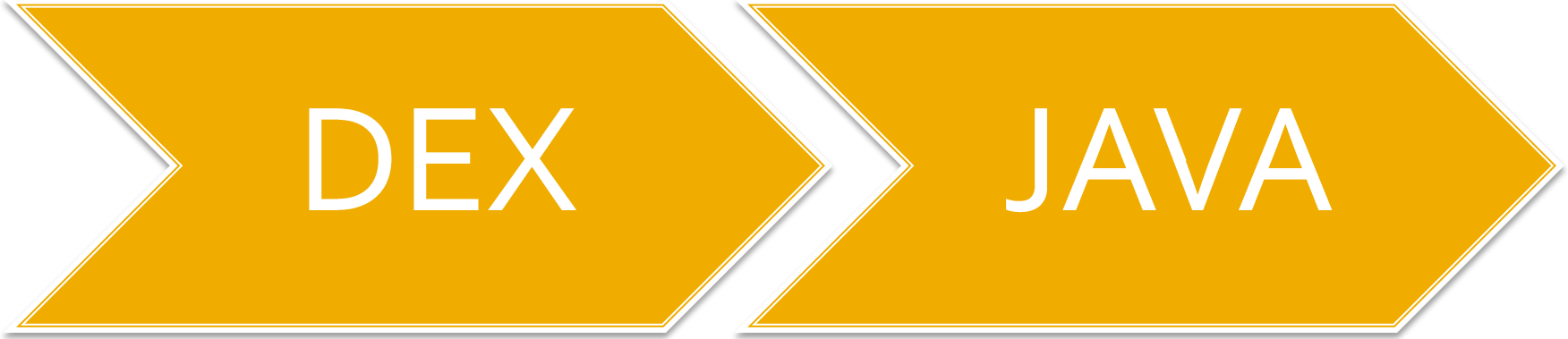
```
santoku@santoku-VirtualBox:/tmp/apk$ unzip test.apk classes.dex
Archive:  test.apk
  inflating: classes.dex
santoku@santoku-VirtualBox:/tmp/apk$ d2j-dex2jar classes.dex
dex2jar classes.dex -> classes-dex2jar.jar
santoku@santoku-VirtualBox:/tmp/apk$ ls -l classes*
-rw-rw-r-- 1 santoku santoku 38520 Dec 19  2011 classes.dex
-rw-rw-r-- 1 santoku santoku 31589 Jan  3 16:27 classes-dex2jar.jar
santoku@santoku-VirtualBox:/tmp/apk$
```

Java Decompilers

- **Jd-gui** - http://jd.benow.ca/
  - Multi platform
  - Closed source
- **JAD** - http://varaneckas.com/jad/
  - Multi platform
  - Closed source
  - Command line
- **Procyon** - https://bitbucket.org/mstrobel/procyon
  - Multi platform (java)
  - Open source (Apache 2)
  - Command line
- Others: **Dare**, **Mocha**, …



Decompiling – Android (Dalvik) decompiler



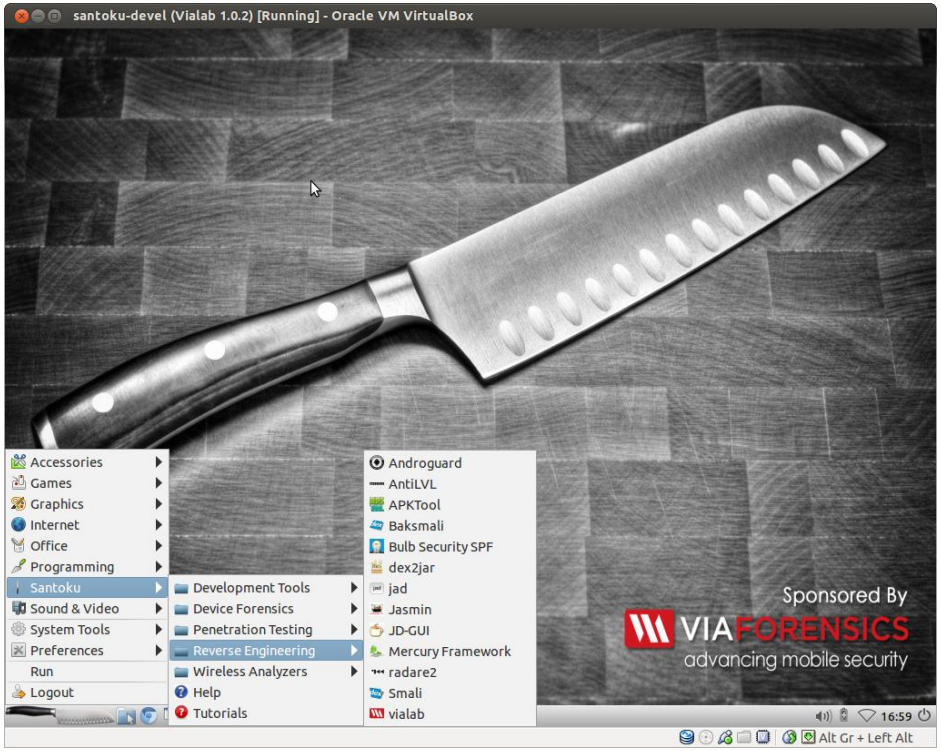
Dalvik Decompilers

- Transforming DEX to JAR looses important metadata that the decompiler could use.
  - Pure Dalvik decompilers skip this step, so they produce better output
- Unfortunately there are not as many choices for Android decompilers as for Java decompilers:
  - Open Source:
    - **Androguard's DAD** - https://code.google.com/p/androguard/
    - **Jadx** - https://github.com/skylot/jadx
  - Commercial:
    - **JEB** - http://www.android-decompiler.com/
  - Others?

Demo – Santoku

Demo – Santoku Linux

- **Santoku Linux -** https://santoku-linux.com/
  - Mobile Forensics
  - Mobile Malware analysis
  - Mobile application assessment



Common app vulnerabilities and FAILs

Common app FAILs

- Not encrypting locally stored data
  - Userdata & sdcard
- Not using SSL connections
  - Or using them without pinning (Certificate pinning & Public key pinning)
- Not protecting app components
  - Custom permissions ("first one wins")
  - Unprotected intents
- Not validating client data
  - Content provider injections
  - Path traversals
- Leaking sensitive data
  - Device logcat, network, memory…

Building in-app security

Building in-app security

- Asses risk of data stored
- Bundle your own crypto libraries
  - SpongyCastle supports AES-GCM & ECC
- Use the KeyStore provider (Android 4.3+)
  - Hardware backed (on some devices)
- Session timeout (app & server side)
  - Clear app data from memory
- Tamper detection
  - Validate signing key

Quick Wins

- **Secure-Preferences** - https://github.com/scottyab/secure-preferences
  - Android Shared preference wrapper than encrypts the keys and values of Shared Preferences
- **SQLCipher** - https://guardianproject.info/code/sqlcipher
  - SQLite extension that provides transparent AES encryption of database files.
- **IOCipher** - https://guardianproject.info/code/iocipher
  - virtual encrypted disk for apps using a clone of the standard java.io API
- **Conceal** - http://facebook.github.io/conceal/
  - Easy to use APIs for fast encryption and authentication of data

Code obfuscation & anti-reversing

- **Proguard**
  - File shrinker, Dex optimizer, Obfuscator, Preverifier
  - Removes unused classes, fields, methods & attributes
  - Renames classes, fields and methods using short names (a, b, c, d,…)
  - Integrated in the Android SDK

Code obfuscation & anti-reversing

- **Dexguard**
  - Comercial version of Proguard
  - Focus on code protection:
    - String encryption
    - Class encryption
    - API hiding
    - UTF16 class names

Code obfuscation & anti-reversing

- Other packers/obfuscators:
  - **APK Protect**: anti-debugging, java and jni obfuscation
  - **HoseDex2Jar:** embeds encrypted DEX into a regular DEX header – see PracticingSafeDex from Tim Strazzere
  - **BangCLE**: uses encrypted DEX, decrypted at runtime by an encrypted ELF and then loaded via class loader
  - **Ijiami**, **Morpher**, **Cryptanium**, etc…

Summary

- Apktool helps extracting & repacking APKs
- Dex2jar converts Dalvik Bytecode to Java Bytecode.
- Santoku Linux has all the tools you need to reverse engineering mobile apps
- Don't do the common app FAILs
- Use the "quick wins" to easily protect your apps

Q&A | Contact | Feedback

- Thanks for listening…

@pof

github.com/poliva

poliva@viaforensics.com