Android reverse engineering: understanding third-party applications

Vicente Aguilera Díaz
OWASP Spain Chapter Leader
Co-founder of Internet Security Auditors
vicente.aguilera@owasp.org
Twitter: @vaguileradiaz
www.vicenteaguileradiaz.com

OWASP EU Tour 2013
June 5, 2013. Bucharest (Romania)
Who I am?

- Co-founder of Internet Security Auditors
- OWASP Spain Chapter Leader
- More info: www.vicenteaguileradiaz.com
Agenda

- Reverse engineering: definition and objectives
- Application analysis workflow
- Malware identification in Android apps
Reverse engineering: definition and objectives

Definition

- Refers to the process of analyzing a system to identify its components and their interrelationships, and create representations of the system in another form or a higher level of abstraction. [1]

Objectives

- The purpose of reverse engineering is not to make changes or to replicate the system under analysis, but to understand how it was built.
Application analysis workflow

Scope of this presentation

Original APK

- Decompress and Dissassemble

Analyze

Rebuild APK

Modify

Modified APK
Application analysis workflow

App Name

SaveAPK
Astro File Manager
Real APK Leecher

APK

apktool

unzip

radare2

AndroidManifest.xml
apktool.yml
/assets
/res
/smali
.smali

apktool

/baksmali

/baksmali

.dex2jar

 classe.dex

.dex2jar

.class

.dex2jar

.java

AXMLPrinter2.jar

Human-readable XML

Static Analysis

Static Analysis

Understand Dexter
grep

Disasm
Debug
Analyze
Manipulate

OWASP
Application analysis workflow

### Static Analysis Tools for Android Apps

<table>
<thead>
<tr>
<th>TOOL</th>
<th>DESCRIPTION</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dexter</td>
<td>Static android application analysis tool</td>
<td><a href="https://dexter.bluebox.com/">https://dexter.bluebox.com/</a></td>
</tr>
<tr>
<td>Androguard</td>
<td>Analysis tool (.dex, .apk, .xml, .arsc)</td>
<td><a href="https://code.google.com/p/androguard/">https://code.google.com/p/androguard/</a></td>
</tr>
<tr>
<td>smali/baksmali</td>
<td>Assembler/disassembler (dex format)</td>
<td><a href="https://code.google.com/p/smali/">https://code.google.com/p/smali/</a></td>
</tr>
<tr>
<td>apktool</td>
<td>Decode/rebuild resources</td>
<td><a href="https://code.google.com/p/android-apktool/">https://code.google.com/p/android-apktool/</a></td>
</tr>
<tr>
<td>JD-GUI</td>
<td>Java decompiler</td>
<td><a href="http://java.decompiler.free.fr/?q=jdgui">http://java.decompiler.free.fr/?q=jdgui</a></td>
</tr>
<tr>
<td>Dedexer</td>
<td>Disassembler tool for DEX files</td>
<td><a href="http://dedexer.sourceforge.net/">http://dedexer.sourceforge.net/</a></td>
</tr>
<tr>
<td>AXMLPrinter2.jar</td>
<td>Prints XML document from binary XML</td>
<td><a href="http://code.google.com/p/android4me/">http://code.google.com/p/android4me/</a></td>
</tr>
<tr>
<td>dex2jar</td>
<td>Analysis tool (.dex and .class files)</td>
<td><a href="https://code.google.com/p/dex2jar/">https://code.google.com/p/dex2jar/</a></td>
</tr>
<tr>
<td>apkinspector</td>
<td>Analysis functions</td>
<td><a href="https://code.google.com/p/apkinspector/">https://code.google.com/p/apkinspector/</a></td>
</tr>
<tr>
<td>Understand</td>
<td>Source code analysis and metrics</td>
<td><a href="http://www.scitools.com/">http://www.scitools.com/</a></td>
</tr>
<tr>
<td>Agnitio</td>
<td>Security code review</td>
<td><a href="http://sourceforge.net/projects/agnitiotool/">http://sourceforge.net/projects/agnitiotool/</a></td>
</tr>
</tbody>
</table>
## Application analysis workflow

### Others (necessary) tools

<table>
<thead>
<tr>
<th>TOOL</th>
<th>DESCRIPTION</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--- emulator</td>
<td>Virtual mobile device</td>
</tr>
<tr>
<td></td>
<td>--- adb</td>
<td>Android debug bridge</td>
</tr>
<tr>
<td>A.R.E.</td>
<td>Android Reverse Engineering VM</td>
<td><a href="https://redmine.honeynet.org/projects/are/wiki">https://redmine.honeynet.org/projects/are/wiki</a></td>
</tr>
</tbody>
</table>
Malware identification in Android apps

- Malware definition
  - Malware is a piece of code which changes the behavior of either the operating system kernel or some security sensitive applications, without a user consent and in such a way that it is then impossible to detect those changes using a documented features of the operating system or the application.[2]
  - A malware is any malicious code or piece of software that is designed to perform functions without the consent of the user.
Malware identification in Android apps

Techniques for introducing malware

- Exploit any vulnerability in the web server hosting the official store
- Use the official store to post apps containing malware
- Install not malicious app that, at some point, install malicious code
- Use alternatives[3] to official stores to post apps containing malware
Malware identification in Android apps

- A practical example
- Some considerations
  - The analyzed app are in the Play Store
  - The published application does not exploit (supposedly) any vulnerability, but can contains malicious code that exploits the user's trust[4]
  - We will only use static analysis
  - We will analyze Java source code
  - We will use the Android Emulator[5]
Malware identification in Android apps

What do we need?

... and motivation!
Malware identification in Android apps

Let's see an example...
Malware identification in Android apps

- Identify a possible malicious application
  - App with unnecessary permissions
    - A wallpaper that requires “SEND SMS MESSAGES”
    - A calculator that requires “DIRECTLY CALL PHONE NUMBERS”
    - ...
  - Google:
    - +"send sms messages" +"wallpaper" +site:“play.google.com”
Malware identification in Android apps

- Identify a possible malicious application
  - Example: “Pipe Mania Droid Lite”

**THIS APPLICATION HAS ACCESS TO THE FOLLOWING:**

**YOUR MESSAGES**

**RECEIVE TEXT MESSAGES (SMS)**
Allows the app to receive and process SMS messages. This means the app could monitor or delete messages sent to your device without showing them to you.

**SEND SMS MESSAGES**
Allows the app to send SMS messages. This may result in unexpected charges. Malicious apps may cost you money by sending messages without your confirmation.

**NETWORK COMMUNICATION**

**FULL NETWORK ACCESS**
Allows the app to create network sockets and use custom network protocols. The browser and other applications provide means to send data to the internet, so this permission is not required to send data to the internet.

**PHONE CALLS**

**READ PHONE STATUS AND IDENTITY**
Allows the app to access the phone features of the device. This permission allows the app to determine the phone number and device IDs, whether a call is active, and the remote number connected by a call.

**STORAGE**

**MODIFY OR DELETE THE CONTENTS OF YOUR USB STORAGE**
Allows the app to write to the USB storage.
Malware identification in Android apps

- Obtaining the APK file
  - Using the SaveAPK tool (requires IO File Manager)
Malware identification in Android apps

- Decompress the APK file
  - unzip Pipe\ Mania\ Droid\ Lite.apk
- Verify the permissions and receivers
  - `java -jar AXMLPrinter2.jar AndroidManifest.xml > out`

```xml
<receiver
  android:name="com.fortumo.android.BillingSMSReceiver"
>
  <intent-filter
    >
    <action
      android:name="android.provider.Telephony.SMS_RECEIVED"
    >
    </action>
  </intent-filter>
</receiver>

<uses-permission
  android:name="android.permission.RECEIVE_SMS"
>
</uses-permission>
<uses-permission
  android:name="android.permission.SEND_SMS"
>
</uses-permission>
```
Malware identification in Android apps

- Convert from Dalvik EXecutable to Java classes
  - `d2j-dex2jar.sh pipe.apk`
- Decompile Java classes and download source code
  - `jd-gui pipe-dex2jar.jar`
Malware identification in Android apps

- Decompress the source code
  - `unzip pipe-dex2jar-src.zip`

- Search sensitive strings
  - `grep -i telephonymanager -r *`

- Analyze the code
  - With tools
  - Manually

- Identifies malicious code
Malware identification in Android apps

■ “Understand” tool
Malware identification in Android apps

- “Dexter” online service
Malware identification in Android apps

■ “virustotal.com” online service
References

References

http://code.google.com/p/androguard/wiki/DatabaseAndroidMalwares
Thank’s!

Vicente Aguilera Díaz
@vaguileradiaz

www.vicenteaguileradiaz.com