HOOKER
A solution to analyze Android markets

Dimitri Kirchner – @Tibapbedoum
Georges Bossert – @Lapeluche

AMOSSYS
PhD candidate AMOSSYS / Supelec
IT security engineer AMOSSYS
Protocole Reverse Engineering
Android
IT security engineer at AMOSSYS since 2010

Android

Informatique de confiance
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Android security model

Ask the user for permissions in order to access phone resources (texts, GPS, etc.)
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Let’s say, I really need this app…

What the application does with its resources? Are resources really used by the application? Are resources used in a legitimate way?
You already have solutions for that
Static versus dynamic analysis tools
Androguard
JD-Core/GUI
Etc.

Hooker: a solution to analyze Android markets
Dynamic analysis
Solution 1: Build a custom Android ROM (Droidbox) to instrument the kernel
Dynamic analysis

Solution 1: Build a custom Android ROM (Droidbox) to instrument the kernel

Solution 2: Modify APK before install (APIMonitor / Fino) to instrument the APK
Dynamic analysis

Solution 1: Build a custom Android ROM (Droidbox) to instrument the kernel
Solution 2: Modify APK before install (APIMonitor / Fino) to instrument the APK
Solution 3: API hooking framework (Substrate / Xposed)
Online scanners
Mix of static and dynamic
Fancy user interface and reports
Error 524
A timeout occurred

Ray ID: 12fa2ab8911204a3

You
Browser
Working

Paris
CloudFlare
Working

www.apk-analyzer.net
Host
Error

Hooker: a solution to analyze Android markets
Analysis are centered on one application

Is it possible to analyze more than one application?

Can you analyze an entire market?
Introducing hooker
What is Hooker

A solution to analyze Android applications

Centralize and aggregate analysis of thousands of different applications
Come in WE'RE OPEN SOURCE
How Hooker works

Microanalysis versus Macroanalysis
How Hooker works

Microanalysis versus Macroanalysis

Analysis on one specific application
How Hooker works

Microanalysis versus Macroanalysis

Analysis on one specific application

Analysis of several applications
Microanalysis overview

Step 1
Static analysis

Step 2
Dynamic analysis

Step 3
Distributed database

Step 4
Data mining

Application
Rule n1: Gather all possible information about the application behavior.
Step 1: Androguard

It just works great
Framework in python

Let us extract basic information about the application

Package name
Permissions
Services
Etc.
Step 2: Substrate

An API hooking framework

Changes behavior of one application, without patches, or specific ROM, or whatever

What you need is:

Root access

Compatible Android version
Substrate

Injects code into Zygote process (father of all processes)
Therefore, injected in all spawned processes
(Similar to Xposed)
Use Substrate to:

Hook access to personal information (read contacts, etc.)

Hook access to specific API (open socket)

Modify return of specific methods (anti-anti-emulation)
Hook PowerManager methods

```java
/**
 * Attach on PowerManager class.
 */

private void attachOnPowerManagerClass() {
    final String className = "android.os.PowerManager";

    Map<String, Integer> methodsToHook = new HashMap<String, Integer>();
    methodsToHook.put("goToSleep", 0);
    methodsToHook.put("isScreenOn", 0);
    methodsToHook.put("newWakeLock", 1);
    methodsToHook.put("reboot", 1);
    methodsToHook.put("userActivity", 1);
    methodsToHook.put("wakeUp", 1);

    try {
        hookMethods(null, className, methodsToHook);
        SubstrateMain.log(new StringBuilder("hooking ").append(className)
                .append(" methods sucessful").toString());
    } catch (HookerInitializationException e) {
        SubstrateMain.log(new StringBuilder("hooking ").append(className)
                .append(" methods has failed").toString(), e);
    }
}
```
Hook PowerManager methods

```java
/**
 * Attach on PowerManager class.
 */

private void attachOnPowerManagerClass() {
    final String className = "android.os.PowerManager";

    Map<String, Integer> methodsToHook = new HashMap<String, Integer>();
    methodsToHook.put("goToSleep", 0);
    methodsToHook.put("isScreenOn", 0);
    methodsToHook.put("newWakeLock", 1);
    methodsToHook.put("reboot", 1);
    methodsToHook.put("userActivity", 1);
    methodsToHook.put("wakeup", 1);

    try {
        hookMethods(null, className, methodsToHook);
        SubstrateMain.log(new StringBuilder("hooking ").append(className) .append(" methods successful").toString());
    } catch (HookerInitializationException e) {
        SubstrateMain.log(new StringBuilder("hooking ").append(className) .append(" methods has failed").toString(), e);
    }
}
Build events in real time

APK

Substrate

API Android

Hooker: a solution to analyze Android markets
Build events in real time

APK

Substrate

Event object

API Android

Collect service

Hooker: a solution to analyze Android markets
Build events in real time

APK
- Substrate

Event object

API Android

Collect service

Unique identifier
Relative and absolute timestamps
Parameters type and value
Return type and value
Intrusive level

Hooker: a solution to analyze Android markets
Intrusive level indicator

Differentiates critical event from normal event
Writing is considered more intrusive than reading
Application doing lots of intrusive events is highlighted
Hooker: a solution to analyze Android markets

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td></td>
<td>javax.crypto.Cipher</td>
</tr>
<tr>
<td>HookerName</td>
<td></td>
<td>Crypto</td>
</tr>
<tr>
<td>InstanceID</td>
<td></td>
<td>1094434584</td>
</tr>
<tr>
<td>IntrusiveLevel</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>MethodName</td>
<td></td>
<td>init</td>
</tr>
<tr>
<td>PackageName</td>
<td></td>
<td>com.melodis.midomiMusicIdentifier</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
<td>{&quot;ParameterType&quot;:&quot;java.lang.Integer&quot;,&quot;ParameterValue&quot;:&quot;1&quot;},</td>
</tr>
</tbody>
</table>
|                |        | {"ParameterType":"javax.crypto.spec.SecretKeySpec","ParameterValue":"SecretKeySpec[algorithm=AES/ECB/PKCS7Padding,key=
|                |        | [101,53,98,49,102,56,52,102,101,49,49,48,52,100,99,52,97,97,102,99,52,99,102,54,99,102,100,102,48,99,50,56]"}
| RelativeTimestamp |    | 53274809                                                            |
| Timestamp      |        | 1397378485227                                                        |
| _id            |        | wttEjbPRTsKJKUmaHc4JSQ                                               |
| _index         |        | hooker_test                                                          |
| _type          |        | event                                                                |
Hooker: a solution to analyze Android markets
<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassName</td>
<td></td>
<td>javax.crypto.Cipher</td>
</tr>
<tr>
<td>PackageName</td>
<td></td>
<td>com.melodis.midomiMusicIdentifier</td>
</tr>
<tr>
<td>RelativeTimestamp</td>
<td></td>
<td>53274809</td>
</tr>
<tr>
<td>Timestamp</td>
<td></td>
<td>1397378485227</td>
</tr>
<tr>
<td>_id</td>
<td></td>
<td>wttEjbrPTsKJKUmaHc4JSQ</td>
</tr>
<tr>
<td>_index</td>
<td></td>
<td>hooker_test</td>
</tr>
<tr>
<td>_type</td>
<td></td>
<td>event</td>
</tr>
</tbody>
</table>

View: Table / JSON / Raw

Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets
Main limitation

White list enumeration

We don't intercept what we don't declare
Main limitation

White list enumeration

We don’t intercept what we don’t declare

Test on malware if we intercept everything we want
Hooker: a solution to analyze Android markets
Hooker: a solution to analyze Android markets

Step 3 and 4: Collects and centralizes results for data mining
Store events in a distributed database

Elastic search

Interact with database

Kibana (front-end)
Hooker: a solution to analyze your Android market

**Example of Kibana web interface**

- **Histogram**
  - View
  - Network (306) • HookerName:SQLi (26) • HookerName:IPC (64) • HookerName:Telephony (26)
  - HookerName:SharedPreferences (168) • HookerName:Crypto (2751) • HookerName:DRM (0) • HookerName:Geolocation (0)
  - HookerName:Accounts (0) • HookerName:ContentsData (82) • HookerName:DynamicCodeLoader (0) • HookerName:FileSystemHooker (802)
  - HookerName:System (12) • HookerName:TaskScheduler (24) • HookerName:MediaRecorder (0) • HookerName:Runtime (26)
  - HookerName:Resources (384) • count per 1s | (4671 hits)

- **Hooker Names**
  - FileSystemHooker 17%
  - Crypto 59%

- **Methods**
  - update (1791) • getinstance (481) • digest (475)
  - getstring (339) • exists (222) • file (166) • getabsolutepath (86)
  - getboolean (84) • getpath (79) • bind (78)
You have to build your own Kibana interface

Basic malware generates 2000 events in 60 seconds
Macroanalysis
Macroanalysis

Automation and parallelization of microanalysis
Macroanalysis

Automation and parallelization of microanalysis

Look for specific patterns in thousands of applications
Macroanalysis

Automation and parallelization of microanalysis

Look for specific patterns in thousands of applications

Post analysis
Macroanalysis

Automation and parallelization of microanalysis

Look for specific patterns in thousands of applications
Automation

**Step 1:** Prepare an Android emulator

**Step 2:** Configure a scenario

- Install
- Execute
- Stimulate
- External stimulation
- Reboot
Automation

**Step 1:** Prepare an Android emulator

**Step 2:** Configure a scenario
- Install
- Execute
- Stimulate
- External stimulation
- Reboot

Hooker: a solution to analyze Android markets
Automation

**Step 1:** Prepare an Android emulator

**Step 2:** Configure a scenario
- Install
- Execute
- Stimulate
- External stimulation
- Reboot
Automation

**Step 1:** Prepare an Android emulator

**Step 2:** Configure a scenario

- Install
- Execute
- Stimulate
- External stimulation
- Reboot

Hooker: a solution to analyze Android markets
Step 3: Run the experiment

$ python hooker_xp.py -c automaticAnalysis.conf

Wait and see
Post-analysis
Python script to query Elasticsearch database
Query what you want to make:
• Statistics
• Highlights
Get thousands of APKs
Google store
Unofficial markets
APK in archives
Get thousands of APKs
Google store
Unofficial markets
APK in archives

What we have tried until now:
1000 apps from SlideMe market in the paper
1000 apps from Google store
Network statistics
Most used Network methods

- getFile
- sendto
- recvfrom
- execute
- getAuthority
- getInputStream
- IOException
- getSettings
- getOutputStream
- URL
- Socket
- openConnection
- getHost
- closeSocket
- getPort
- close
- connect
- getProtocol
- setCertificate

Number of applications
Internet permissions
477 apps asking for internet permissions
404 have been found using it
Domains most accessed

- www.google-analytics.com
- mm.admob.com
- googleads.g.doubleclick.net
- www.google.com
- ade.wooboo.com.cn
- secure.gameloft.com

Number of applications
Domains most accessed

- mm.admob.com: 33
- googleads.g.doubleclick.net: 20
- www.google-analytics.com: 15
- www.google.com: 10
- ade.wooboo.com.cn: 8
- secure.gameloft.com: 5

Advertisements
Port number accessed by applications

Number of applications

Port number

5122
305
1130
5220
443
80
Hooker: a solution to analyze your Android market

Port number accessed by applications

- Port number: 80
- Number of applications: 150

Noknok trojan?
Wanna find some vulnerable apps?
Webview and 
addJavascriptInterface
Webview and
addJavascriptInterface

Interface to call Java from Javascript
Webview and addJavascriptInterface

Interface to call Java from Javascript

Remote code execution
On 1000 applications from Google store
23 apps using `addJavascriptInterface` method
Crypto statistics
Hooker: a solution to analyze your Android market
Use of cipher functions

- DESede
- PBEwithMD5andDES
- AES/ECB/NoPadding
- AES/ECB/PKCS5Padding
- AES/CBC/NoPadding
- AES/CBC/PKCS7Padding
- Blowfish
- DES
- AES/CBC/PKCS5Padding
- DES/CBC/PKCS5Padding
- AES
- PBEWithSHA256And256BitAES-CBC-BC
Bitcoin miners
«Several apps from the GPlay are infected by crypto miners.»
Several apps from the GPlay are infected by crypto miners

Crypto hashing abuses
File statistics
Files accessed by application other than their /data/

- /data/misc/keychain/cacerts-added
- /vendor/lib/libmedia_jni.so
- /data/misc/keychain/cacerts-removed
- /vendor/lib/libsoundpool.so
- /mnt/sdcard
- /system/etc/security/cacerts
- /sdcard
- /system/lib/libsoundpool.so
- /proc/meminfo
- /
- /system/lib/libmedia_jni.so
- /proc/cpuinfo
Files accessed by application other than their /data/

- /data/misc/keychain/cacerts-added
- /vendor/lib/libmedia_jni.so
- /data/misc/keychain/cacerts-removed
- /vendor/lib/libsoundpool.so
- /mnt/sdcard
- /system/etc/security/cacerts
- /sdcard
- /system/lib/libsoundpool.so
- /proc/meminfo
- /
- /system/lib/libmedia_jni.so
- /proc/cpuinfo

Number of applications
Certificates

Files accessed by application other than their /data/

- /data/misc/keychain/cacerts-added
- /vendor/lib/libmedia_jni.so
- /data/misc/keychain/cacerts-removed
- /vendor/lib/libsoundpool.so
- /mnt/sdcard
- /system/etc/security/cacerts
- /sdcard
- /system/lib/libsoundpool.so
- /proc/meminfo
- /proc/cpuinfo

Number of applications

Hooker: a solution to analyze your Android market
Hooker: a solution to analyze your Android market

File accesses are illustrating application behavior...

<table>
<thead>
<tr>
<th>Time</th>
<th>Process/Thread/Level</th>
<th>Log Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>543</td>
<td>[MainProcess/MainThread/INFO] 31798</td>
<td>7c358ec3a35f2eb3034f12ba3e76b613 accessing files:</td>
</tr>
<tr>
<td>544</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/WidgetPreview.apk</td>
</tr>
<tr>
<td>545</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/WidgetPreview.apk@classes.dex</td>
</tr>
<tr>
<td>546</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/SmokeTestApp.apk</td>
</tr>
<tr>
<td>547</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/SmokeTestApp.apk@classes.dex</td>
</tr>
<tr>
<td>548</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/SmokeTest.apk</td>
</tr>
<tr>
<td>549</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/SmokeTest.apk@classes.dex</td>
</tr>
<tr>
<td>550</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/com.amossys.hooker-2.apk</td>
</tr>
<tr>
<td>551</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/com.amossys.hooker-2.apk@classes.dex</td>
</tr>
<tr>
<td>552</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/com.amossys.hooker.generatecontacts-1.apk@classes.dex</td>
</tr>
<tr>
<td>553</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/GestureBuilder.apk</td>
</tr>
<tr>
<td>554</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/GestureBuilder.apk@classes.dex</td>
</tr>
<tr>
<td>555</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/com.amossys.hooker.generatecontacts-1.apk</td>
</tr>
<tr>
<td>556</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/SoftKeyboard.apk</td>
</tr>
<tr>
<td>557</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/SoftKeyboard.apk@classes.dex</td>
</tr>
<tr>
<td>558</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/ApiDemos.apk</td>
</tr>
<tr>
<td>559</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/ApiDemos.apk@classes.dex</td>
</tr>
<tr>
<td>560</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/com.saurik.substrate-1.apk@classes.dex</td>
</tr>
<tr>
<td>561</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/CubeLiveWallpapers.apk</td>
</tr>
<tr>
<td>562</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/CubeLiveWallpapers.apk@classes.dex</td>
</tr>
<tr>
<td>563</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/com.noshufou.android.su-1.apk</td>
</tr>
<tr>
<td>564</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/dalvik-cache/data@app/com.noshufou.android.su-1.apk@classes.dex</td>
</tr>
<tr>
<td>565</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/data/app/com.saurik.substrate-1.apk</td>
</tr>
<tr>
<td>566</td>
<td>[MainProcess/MainThread/INFO] 31799</td>
<td>/sdcard/backups/apps</td>
</tr>
</tbody>
</table>
Backup app

Hooker: a solution to analyze your Android market
How to be sure to find an sdcard...
you have to trust this app...

```
729 [MainProcess/MainThread/INFO] 31469: /system/xbin/iptables
730 [MainProcess/MainThread/INFO] 31469: /system/xbin/su
731 [MainProcess/MainThread/INFO] 31469: /system/bin/iptables
```
That's weird right?
That's weird right?

Is this app legitimate?
Hooker has a lot more capabilities
You chose to extract what you want
Highlight **weaknesses** in application
Highlight **malwares** within thousands of applications
Highlight **WTF** behavior on your system
Highlight weaknesses in application
Highlight malwares within thousands of applications
Highlight WTF behavior on your system

Give it a try, play hooker now:
https://github.com/AndroidHooker
Questions

Play hooker now:
https://github.com/AndroidHooker