

CobraDroid

HOOKING ANDROID APPLICATIONS

Jake Valletta
BruCON 2013

About Me

- Consultant at Mandiant
- Pen-testing, IR, forensics, application security
 - Strong interests in mobile security
- Mobile security blog and research: “The Cobra Den”
 - <http://blog.thecobraden.com/>
 - <http://www.thecobraden.com/>
- @jake_valletta

Agenda

- Background & Overview
- CobraDroid Features
- Demo
- Future Plans
- Questions & Answers



Background & Overview

Current Situation – Background

- People want/need to analyze Android applications
 - Companies pay to be told they are “safe”
 - Analyzing malware
 - General curiosity (*why is Angry Birds asking to use my camera?*)

Current Situation – Static Analysis?

- Lots of tools!
 - Smali/Baksmali
 - Dex2jar
 - Apktool
 - Dexter by BlueBox
 - IDA Pro
- Lots of information on how to tear applications apart...
 - ...And modify and repackage!

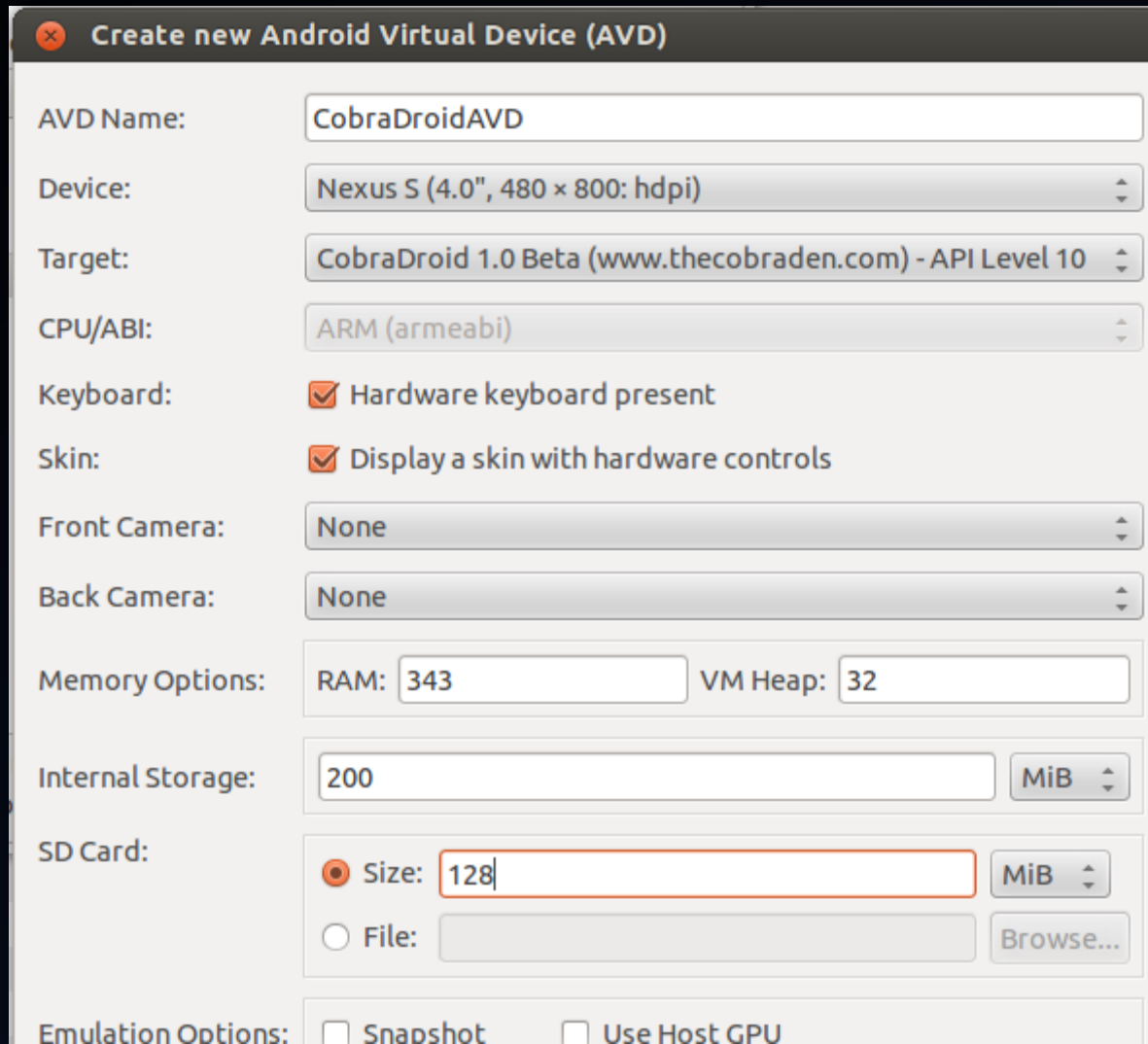
Current Situation – Dynamic Analysis?

- Less common
 - “AppUse” by AppSecLabs (closed-source)
- There are plenty of services that will analyze your application
 - Upload to website, get results
 - NOT ideal for client related work
 - “Blackbox”

Goals of CobraDroid

- Create a free and open dynamic analysis platform
 - Needs to be easy to install, setup, and use
- Give the tester as much control and visibility as possible
 - Make their job easier and successful
- Learn about Android internals 😊

Using CobraDroid



Create new Android Virtual Device (AVD)

AVD Name: CobraDroidAVD

Device: Nexus S (4.0", 480 × 800: hdpi)

Target: CobraDroid 1.0 Beta (www.thecobraden.com) - API Level 10

CPU/ABI: ARM (armeabi)

Keyboard: ☒ Hardware keyboard present

Skin: ☒ Display a skin with hardware controls

Front Camera: None

Back Camera: None

Memory Options: RAM: 343 VM Heap: 32

Internal Storage: 200 MiB

SD Card: ☒ Size: 128 MiB ☐ File: Browse...

Emulation Options: ☐ Snapshot ☐ Use Host GPU

- Setup Android SDK
- Download archive from my website
- Unzip to “add-ons” directory (SDK)
- Create new AVD



CobraDroid Features

What is CobraDroid?

- Modified Android build for the emulator
 - QEMU emulating ARM code
 - Android 2.3.7 (“GingerBread”)
- Modified from the lowest point up
 - Kernel
 - User-space libraries + tools
 - Dalvik VM
 - Android applications

Updated Kernel (CobraKernel)

- At the time of development, latest “Goldfish” kernel was 2.6.29
 - “kernel.org” publish date of April 13, 2008
 - Default kernel with Android 1.5 “Donut” (released Sept 19, 2009)
- Updated to 2.6.36
 - Default kernel with Android 3.0 “HoneyComb” (released Feb 22, 2011)
- More powerful configuration
 - Full netfilters
 - Loadable kernel modules

Bash & BusyBox

- Android 2.3 shell is terrible. Terrible.
 - No autocomplete
 - No coloring
 - No pipes
- Lack of tools/utilities
 - No editors
 - No [insert your favorite Unix tool]

Bash & BusyBox

```
root@android-assessment:/home/analyst# adb shell
CobraDroid / # uname -a
Linux localhost 2.6.36-CobraKernel #102 Mon Jul 29 13:38:48 EDT 2013 armv5tejl GNU/Linux
CobraDroid / # ls -la /
drwxr-xr-x 13 root root 0 Sep 17 19:55 .
drwxr-xr-x 13 root root 0 Sep 17 19:55 ..
drwxr-xr-x 3 root root 0 Sep 17 19:55 acct
drwxrwx--- 1 system cache 2048 Sep 17 19:55 cache
dr-x----- 2 root root 0 Sep 17 19:55 config
lrwxrwxrwx 1 root root 17 Sep 17 19:55 d -> /sys/kernel/debug
drwxrwx--x 1 system system 2048 Sep 17 19:56 data
-rw-r--r-- 1 root root 118 Dec 31 1969 default.prop
drwxr-xr-x 10 root root 2120 Sep 17 19:56 dev
lrwxrwxrwx 1 root root 11 Sep 17 19:55 etc -> /system/etc
-rwxr-x--- 1 root root 94168 Dec 31 1969 init
-rwxr-x--- 1 root root 1731 Dec 31 1969 init.goldfish.rc
-rwxr-x--- 1 root root 13827 Dec 31 1969 init.rc
drwxrwxr-x 6 root system 0 Sep 17 19:55 mnt
dr-xr-xr-x 77 root root 0 Dec 31 1969 proc
drwx----- 2 root root 0 Jul 23 18:55 root
drwxr-x--- 2 root root 0 Dec 31 1969 sbin
lrwxrwxrwx 1 root root 11 Sep 17 19:55 sdcard -> /mnt/sdcard
drwxr-xr-x 12 root root 0 Sep 17 19:55 sys
drwxr-xr-x 1 root root 2048 Sep 15 17:44 system
-rw-r--r-- 1 root root 0 Dec 31 1969 ueventd.goldfish.rc
-rw-r--r-- 1 root root 3882 Dec 31 1969 ueventd.rc
lrwxrwxrwx 1 root root 14 Sep 17 19:55 vendor -> /system/vendor
CobraDroid / #
```

LiME Forensics

- Linux Memory Extractor by Joe Sylve (504ensics)
 - <http://code.google.com/p/lime-forensics/>
- Allows for live memory acquisition via Loadable Kernel Module
 - Open saved files with Volatility or Dalvik Inspector
- Modified to fit CobraDroid as device driver + user-space API
 - <https://github.com/jakev/lime-forensics-jakev>

LiME Forensics

- “lime” command line utility
 - Links against “liblime.so”
- “android.jakev.Lime” class for Android applications
 - NOT SAFE - Currently implementing safer solution
 - Gives Android application access to kernel driver

```
CobraDroid / # lime -d/mnt/sdcard/memory.dump -fraw
Disk mode selected: /mnt/sdcard/memory.dump
Output format: raw
About to dump memory to disk...
[find] no, I don't like you. com.jakev.testing.TestingActivity.snakeTestCall
```


Editable Radio & Device Identifiers

- Lets you make the phone look like anything you want!
- Helps with application whitelisting/blacklisting
 - Is this a Vodafone? Telefónica? Is it a Nokia? Motorola?
- Previously very tedious to change on emulator
 - Radio properties: Modify “emulator-arm” binary
 - Device properties: Modify :“/etc/build.prop” and reconstruct the “system.img”

Editable Radio & Device Identifiers

- Re-written “TelephonyManager” class
 - Queries a custom file instead
- Removed “android.os.Build” class initialization in Zygote
 - Hooked “SystemProperties” class
 - Queries a custom file instead

Editable Radio & Device Identifiers

Device ID Control

Set MDN

15555215554

Set VoiceMail Number

+15552175049

Set Device ID (IMEI/MEID)

0000000000000000

Set Subscriber ID (IMSI)

3102600000000000

Set SIM Card Serial

89014103211118510720

Update Values

Custom Build Property Editor

Add New Item

dalvik.vm.stack-trace-file

/data/anr/traces.txt

ro.product.manufacturer

CobraDenSec

ro.product.locale.region

US

ro.build.date

Sat Aug 31 13:32:05 EDT 2013

ro.build.version.release

2.3.7

ro.product.model

CD001

ro.build.id

GWK74

ro.build.fingerprint

generic/CobraDroid/goldfish:2.3.7/
GWK74/eng.root.20130831.133103:eng/
test-keys

Custom Build Property Editor

Add New Item

dalvik.vm.stack-trace-file

/data/anr/traces.txt

ro.product.manufacturer

CobraDenSec

ro.product.locale.region

US

ro.build.date

Sat Aug 31 13:32:05 EDT 2013

ro.build.version.release

2.3.7

ro.product.model

iPhone 5s

ro.build.id

GWK74

ro.build.fingerprint

generic/CobraDroid/goldfish:2.3.7/
GWK74/eng.root.20130831.133103:eng/
test-keys

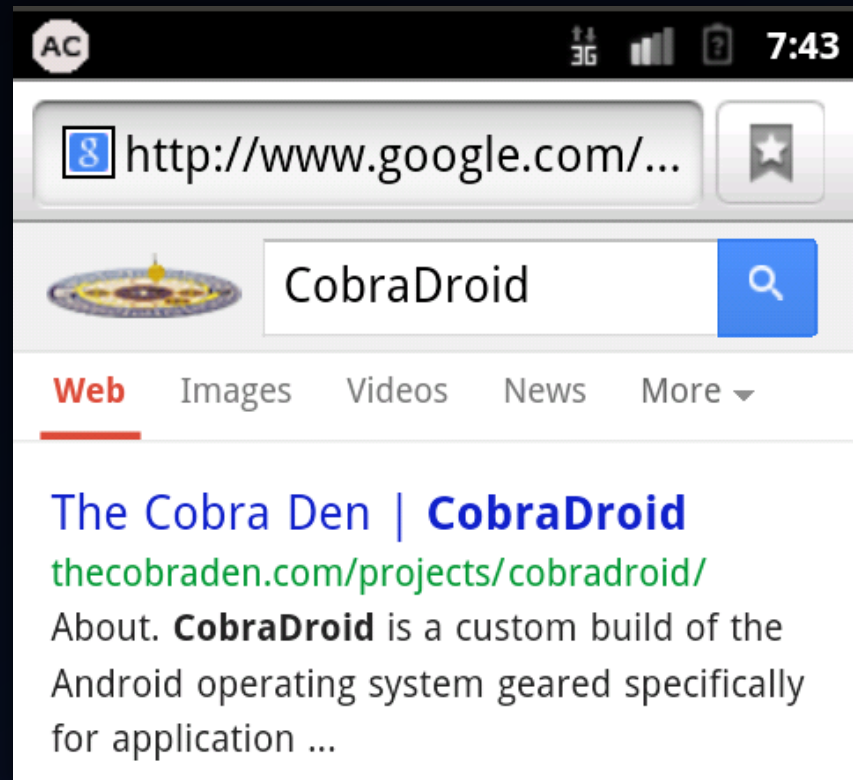
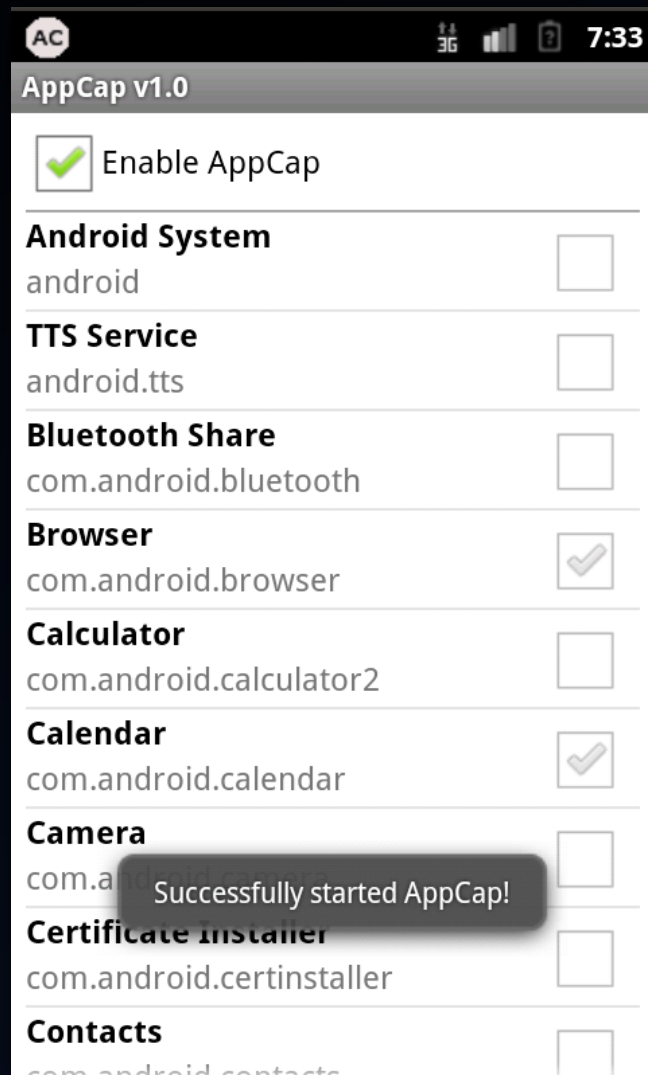
SSL Validation Bypass

- Allows you to man-in-the-middle any SSL connection
 - Disables certificate pinning and CA validation silently
- Re-written constructors and getter/setters
- Works for all default SSL libraries on Android 2.3
 - `HttpsURLConnection` (core.jar)
 - `DefaultHttpClient` (ext.jar)
 - `SSLConnectionFactory` (ext.jar)

Application Specific Packet Capture

- *Show me only traffic for application X (and application Y)*
 - Focus on only the traffic you actually care about
- Uses Custom “iptables” rules to redirect traffic
- View in Wireshark afterwards
 - Tested on 1.8.5 Stable, 1.11.0 Dev. (incompatible with older versions)

Application Specific Packet Capture



```
CobraDroid /mnt/sdcard # ls -l
d---rwxr-x   2 system  sdcard_r    2048 Sep 18 18:32 LOST.DIR
----rwxr-x   1 system  sdcard_r   641024 Sep 18 19:45 appcap-20130918_193754.pcap
CobraDroid /mnt/sdcard #
```

Method Hooking

- CobraDroid uses it to alert on method calls
 - Much more to come
- Could have an entire 45 minute talk on hooking the DVM
 - I'm going to try and do it in about 7 😊
- **TL;DR – Instrumenting method byte-code during Class loading**

Method Hooking

- Configuration file: “/etc/hooks.conf”

```
# Prototype Hook Configuration File
# v0.2

# Our System Hook Section
.sys

# We first select our class of interest, then methods.
android.telephony.TelephonyManager
    getDeviceId @Alert "An application accessed your device ID!"

android.os.Environment
    getExternalStorageDirectory @Alert

.end

#Our Application Hook Section
.app

# An Application Hook
com.jakev.testing.TestingActivity
    snakeTestCall @Alert
    nzkds @Alert "Obfuscated method is accessing your contacts!"
.end
```


Method Hooking

- Configuration file: “/etc/hooks.conf”

```
# Prototype Hook Configuration File  
# v0.2
```

```
# Our System Hook Section
```

```
.sys
```

```
# We first select our class of interest, then methods.
```

```
android.telephony.TelephonyManager
```

```
    getDeviceId @Alert "An application accessed your device ID!"
```

```
android.os.Environment
```

```
    getExternalStorageDirectory @Alert
```

```
.end
```

System JARs

```
#Our Application Hook Section
```

```
.app
```

```
# An Application Hook
```

```
com.jakev.testing.TestingActivity
```

```
    snakeTestCall @Alert
```

```
    nzkds @Alert "Obfuscated method is accessing your contacts!"
```

```
.end
```

Application APKs

Method Hooking

- Configuration file: “/etc/hooks.conf”

```
# Prototype Hook Configuration File
# v0.2
```

```
# Our System Hook Section
```

```
.sys
```

```
# We first select our class of interest, then methods
```

```
android.telephony.TelephonyManager ←
    deviceId @Alert "An application accessed your device ID!"
```

```
android.os.Environment
    getExternalStorageDirectory @Alert
```

```
.end
```

System JARs

Class

Message

Action

Method

```
#Our Application Hook Section
```

```
.app
```

```
# An Application Hook
```

```
com.jakev.testing.TestingActivity
    snakeTestCall @Alert
```

```
    nzkds @Alert "Obfuscated method is accessing your contacts!"
```

```
.end
```

Application APKs

Method Hooking

- It's magic! (Right?)

```
root@android-assessment:/home/analyst# adb logcat -b security
D/EventNotifier( 575): [com.jakev.testing] An application accessed your device ID! "android.telephony.TelephonyManager.getId()"
D/EventNotifier( 575): [com.jakev.testing] Method Call Alert: "android.os.Environment.getExternalStorageDirectory()"
D/EventNotifier( 575): [com.jakev.testing] Method Call Alert: "com.jakev.testing.TestingActivity.snakeTestCall()"
D/EventNotifier( 575): [com.jakev.testing] Obfuscated method is accessing your contacts! "com.jakev.testing.TestingActivity.nzkds()"
^C
root@android-assessment:/home/analyst#
```

Hook Step #1 – DVM Startup

- Read configuration file and parse hooks into global DVM memory
 - Utilize the “gDvm” variable (DvmGlobals struct)
- For each JAR/DEX file, over-allocate strings, methods, etc. based on configuration
 - Modify calloc() calls when initializing “pDvmDex” (DvmDex struct)
 - Structure used to hold resolved classes, methods, etc.

Hook Step #2 – Class/Method Loading

- Read global memory to determine if loaded class and method should be hooked
- For the given method, allocate n bytes for new DexCode struct
 - The original DexCode struct is read-only mapped directly from the DEX file

“DexCode” Structure

Name	Format
registers_size	u2
ins_size	u2
outs_size	u2
tries_size	u2
debug_info_off	u4
insns_size	u4
insns	u2[insns_size]
padding	u2
tries	try_item[tries_size]
handlers	encoded_catch_handler_list

- Contains all declaration details for a method

“DexCode” Structure

Name	Format
registers_size	u2
ins_size	u2
outs_size	u2
tries_size	u2
debug_info_off	u4
insns_size	u4
insns	u2[insns_size]
padding	u2
tries	try_item[tries_size]
handlers	encoded_catch_handler_list

- Contains all declaration details for a method

```
1202                                |0000: const/4 v2, #int 0 // #0
7010 c555 0300                    |0001: invoke-direct {v3}, Landroid/telephony/TelephonyManager;.getSub
0c01                              |0004: move-result-object v1
7210 5c8b 0100                    |0005: invoke-interface {v1}, Lcom/android/internal/telephony/IPhoneSu
0c01                              |0008: move-result-object v1
1101                              |0009: return-object v1
0d01                              |000a: move-exception v1
0710                              |000b: move-object v0, v1
0721                              |000c: move-object v1, v2
```

- “insns” is what we actually want to modify!
 - Add new instructions to do X
- Need to repair structure after

Hook Step #2 – Class/Method Loading

- Add new instructions to “insns”
 - In this case, we call: `Landroid/jakev/EventNotifier;.notifyEvent();`
 - Responsible for printing to logs
 - Optionally add our payload message
- Re-align the remaining DexCode structure
 - Repair “tries”
 - Repair “handlers”

Hook Step #3 – Resolving

- Resolving occurs at runtime, when the DVM must determine what code to run and where it is located
 - Log.d("here", "i am a snake");

```
1337: com.android.systemui | [0006f8] com.jakev.testing.TestingActivity.snakeTestCall:()V
1a00 3300 | 0000: const-string v0, "i am a snake" // string@0033
1a01 4200 | 0002: const-string v1, "this string is hidden!" // string@0042
1a02 3200 | 0004: const-string v2, "here" // string@0032
7120 0400 0200 | 0006: invoke-static {v2, v0}, Landroid/util/Log;
0e00 | .d:(Ljava/lang/String;Ljava/lang/String;)I // method@0004
| 0009: return-void
```

In our app's DEX file

In another DEX file!

Hook Step #3 – Resolving

- **Question:** How do we call a method or use a string that a DexFile struct does not know about?
- **Answer:** Instrument the code with an index beyond the max, then add checks to `dvm.*Resolver()` function calls!
 - i.e. attempting to resolve string 33 out 32
 - Usually this indicates an error condition

Additional Packages

- ProxyDroid
 - Makes it painless to proxy traffic on the emulator
- Superuser/“su”
 - Provides root level access to the device
- Drozer
 - Allows you to assume the role of an Android application at a command line
- EmuCoreTools
 - Front-end interface to CobraDroid features



Demo!

Future Research & Plans

- Move to Ice Cream Sandwich (4.0.0+)
- Expand hooking capabilities
 - Add “payload” action handler
- More “man in the middle” capabilities
 - SQL database queries
 - Intents (broadcast & directed)

Getting More Information

- Check my website & blog for updates, technical material, etc.
 - <http://www.thecobraden.com>
 - <http://blog.thecobraden.com>
- Getting CobraDroid (beta)
 - <http://www.thecobraden.com/projects/cobradroid>
 - <https://github.com/jakev/CobraDroidBeta> (source)

Questions & Answers