Exploiting the Bells and Whistles

Uncovering OEM Vulnerabilities in Android

Jake Valletta

May 18, 2014

Who Am I

- Consultant at Mandiant (er, FireEye)
- Mobile security research and tool development
  - www.thecobraden.com/projects/cobradroid/
  - github.com/jakev/
- @jake_valletta

This talk is NOT…

- An audit of the Android Open Source Project (AOSP)
- An introduction to Android assessment tools
- How to write ARM exploits

This talk is…

- How to determine what manufactures (OEMs) and carriers add and change in the AOSP
- How a malicious user can exploit poorly implemented changes and features
- An exploration of Android platform security

Motivations

- No “primer” on device testing
- No (free) tools for device testing
- Answer the question: Someone hands you a phone – Where are the vulnerabilities?

Motivations

- No “primer” on device testing
- No (free) tools for device testing
- Answer the question: Someone hands you a phone – Where are the vulnerabilities?
  - Where and what to look for
  - What tools to use

Example Vulnerabilities

- Information disclosure
  - Can a malicious application or user "pillage" system or personal data?
- Privilege escalation
  - Can a malicious application or user escalate their privileges on the device?
- Denial of service
  - Can a malicious application cause denial of service like conditions to a device?

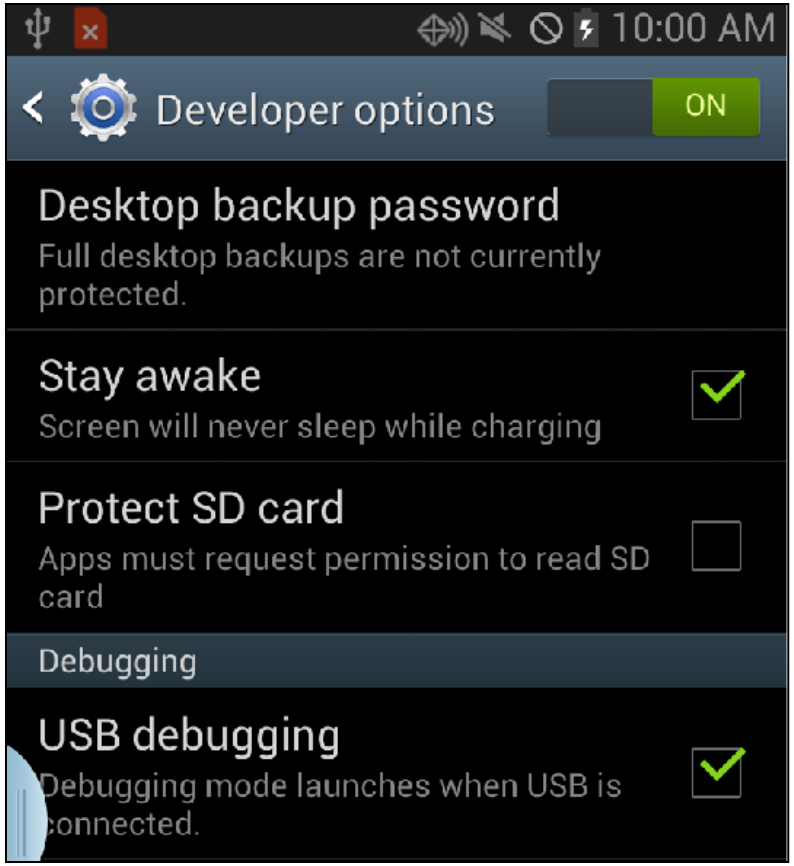
OEM Changes & Additions

Where and What?

Setup

- Physical access + USB Debugging (“adb”)
- No root access

```
root@android-assessment:/# adb devices
List of devices attached
42f70a11c9019fe9        device
```



Application Components

- Activity
  - UI, visual
- Service
  - Background tasks
- Content Provider
  - Abstraction for databases or information sharing via IPC
- Broadcast Receiver
  - Receivers of IPC
- Native library

Application Components

- Can be exported (callable by others)
  - Explicitly
  - Implicitly
    - Debuggable app or "<intent-filter>" presence
- Be careful what you export!
  - Always use permissions

Application Permissions

- Defined by applications
  - Other application components "use" these permissions
  - The Android "core" defines 100+ permissions
- Applied to components
- Different levels of protection
  - normal
  - dangerous
  - systemOrSignature
  - system

Exposed Activities

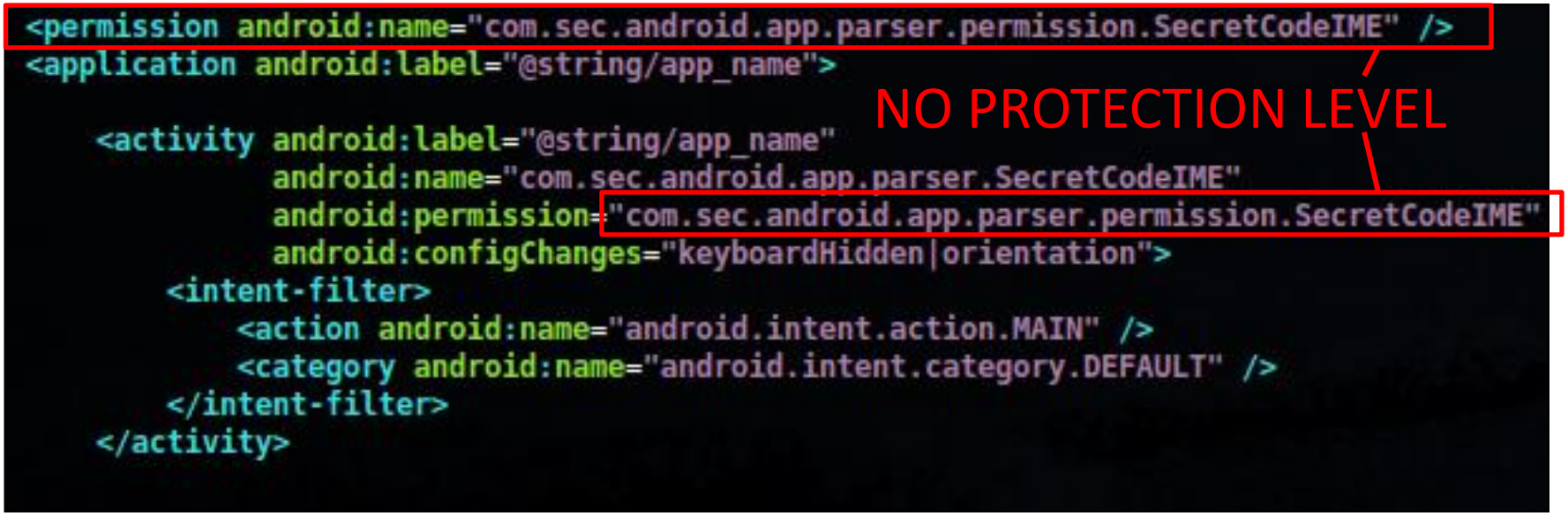
- Usually less critical (still an issue)
- Debugging screens, “hidden” menus, etc.

```
<permission android:name="com.sec.android.app.parser.permission.SecretCodeIME" />
<application android:label="@string/app_name">

    <activity android:label="@string/app_name"
              android:name="com.sec.android.app.parser.SecretCodeIME"
              android:permission="com.sec.android.app.parser.permission.SecretCodeIME"
              android:configChanges="keyboardHidden|orientation">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
```

Exposed Activities

- Usually less critical (still an issue)
- Debugging screens, “hidden” menus, etc.




```
<permission android:name="com.sec.android.app.parser.permission.SecretCodeIME" />
<application android:label="@string/app_name">

    <activity android:label="@string/app_name"
              android:name="com.sec.android.app.parser.SecretCodeIME"
              android:permission="com.sec.android.app.parser.permission.SecretCodeIME"
              android:configChanges="keyboardHidden|orientation">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
```

Exposed Services

- Authenticator services
- Other sensitive actions?

```
<service android:name="com.android.systemui.SystemUIService"
         android:exported="true" />

<service android:name="com.android.systemui.screenshot.TakeScreenshotService"
         android:exported="false"
         android:process=":screenshot" />
```

Exposed Providers

- Databases with sensitive information
  - Wrong permissions
  - No permissions (wut)

```
<provider android:name="ParentalControlSettingsDBHelper"
          android:writePermission="android.permission.WRITE_SETTINGS"
          android:multiprocess="false"
          android:authorities="parentalcontrol"
          android:initOrder="100" />
```

Exposed Providers

- Databases with sensitive information
  - Wrong permissions
  - No permissions (wut)

“Dangerous” Protection Level

```
<provider android:name="ParentalControlSettingsDBHelper"
          android:writePermission="android.permission.WRITE_SETTINGS"
          android:multiprocess="false"
          android:authorities="parentalcontrol"
          android:initOrder="100" />
```

SECRET_CODE Receivers

- Receiver with special data/action Intent filter
- "Backdoor" access to application

```
<receiver android:name="SecKeyStringBroadcastReceiver"
          android:permission="com.sec.android.app.servicemodeapp.permission.KEYSTRING">
    <intent-filter>
        <action android:name="android.provider.Telephony.SECRET_CODE" />
        <data android:scheme="android_secret_code" android:host="197328640" />
        <data android:scheme="android_secret_code" android:host="27663368378" />
        <data android:scheme="android_secret_code" android:host="2684" />
        <data android:scheme="android_secret_code" android:host="0011" />
        <data android:scheme="android_secret_code" android:host="123456" />
        <data android:scheme="android_secret_code" android:host="22553767" />
        <data android:scheme="android_secret_code" android:host="32489" />
        <data android:scheme="android_secret_code" android:host="2580" />
        <data android:scheme="android_secret_code" android:host="9090" />
        <data android:scheme="android_secret_code" android:host="4238378" />
        <data android:scheme="android_secret_code" android:host="745" />
        <data android:scheme="android_secret_code" android:host="66336" />
        <data android:scheme="android_secret_code" android:host="746" />
        <data android:scheme="android_secret_code" android:host="2263" />
        <data android:scheme="android_secret_code" android:host="1575" />
```

Android Frameworks

- Installed to /system/framework/
- Programming APIs, resources
- Loaded into Zygote VM at startup
  - $BOOTCLASSPATH variable

```
root@android-assessment:/# adb shell set |grep BOOTCLASS
BOOTCLASSPATH=/system/framework/core.jar:/system/framework/conscrypt.jar:/system/framewo
rk/okhttp.jar:/system/framework/core-junit.jar:/system/framework/bouncycastle.jar:/syste
m/framework/ext.jar:/system/framework/framework.jar:/system/framework/framework2.jar:/sy
stem/framework/telephony-common.jar:/system/framework/voip-common.jar:/system/framework/
mms-common.jar:/system/framework/android.policy.jar:/system/framework/services.jar:/syst
em/framework/apache-xml.jar:/system/framework/webviewchromium.jar:/system/framework/sec_
edm.jar:/system/framework/seccamera.jar:/system/framework/scrollpause.jar:/system/framew
ork/stayrotation.jar:/system/framework/smartfaceservice.jar:/system/framework/secocsp.ja
r:/system/framework/commonimsinterface.jar:/system/framework/TmoWfcUtils.jar:/system/fra
mework/qcmediaplayer.jar:/system/framework/WfdCommon.jar:/system/framework/oem-services.
jar:/system/framework/org.codeaurora.Performance.jar
```

Android Frameworks

- Need to rebuild “android.jar” to use new APIs in Eclipse
  - Usually need to write in DEX/Smali ☹

```
root@android-assessment:/x# cat ./*|grep -E "\.method"
.method static constructor <clinit>()V
.method public constructor <init>()V
.method public static IDME_read(Landroid/content/Context;Ljava/lang/String;)Ljava/lang/String;
.method public static IDME_write(Ljava/lang/String;Ljava/lang/String;)Z
.method private static native readIDME(Ljava/lang/String;)Ljava/lang/String;
.method public static readSecret(Landroid/content/Context;)Ljava/lang/String;
.method private static native writeIDME(Ljava/lang/String;Ljava/lang/String;)I
root@android-assessment:/x#
```

Android Others

- Android System Service
- /system/permissions/platform.xml
  - Permission to Group ID Mappings
    - Example: “android.permission.INTERNET” → inet
  - Additional permissions assigned to group
    - Example: Give “shell” permission “android.permission.SET_DEBUG_APP”

System Log Buffers

- Located at /dev/log/
- Android provides standard logging capabilities
  - Log.d("MyApp", "CarolinaCon Rulz");
  - events, main, radio, system

```
root@android-assessment:/# adb shell ls -l /dev/log
crw-rw--w- root     log        10,  45 2014-03-28 14:37 amazon_main
crw-rw-rw- root     log        10,  49 2014-03-28 14:37 events
crw-rw-rw- root     log        10,  50 2014-03-28 14:37 main
crw-rw--w- root     log        10,  46 2014-03-28 14:37 metrics
crw-rw-rw- root     log        10,  48 2014-03-28 14:37 radio
crw-rw-rw- root     log        10,  47 2014-03-28 14:37 system
```

System Log Buffers

- Located at /dev/log/
- Android provides standard logging capabilities
  - Log.d("MyApp", "CarolinaCon Rulz");
  - events, main, radio, system

```
root@android-assessment:/# adb shell ls -l /dev/log
crw-rw--w- root     log        10,  45 2014-03-28 14:37 amazon_main
crw-rw-rw- root     log        10,  49 2014-03-28 14:37 events
crw-rw-rw- root     log        10,  50 2014-03-28 14:37 main
crw-rw--w- root     log        10,  46 2014-03-28 14:37 metrics
crw-rw-rw- root     log        10,  48 2014-03-28 14:37 radio
crw-rw-rw- root     log        10,  47 2014-03-28 14:37 system
```

System Binaries

- Can be accessed from command line or from Android app
- Debugging and testing functionality

```
root@android-assessment:/DevTesting/C[redacted]/oem-bins# ls
apaclient               fmfactorytest          mm-jpeg-enc-test       regdbdump
blkid                   fmfactorytestserver    mm-qjpeg-dec-test      resize2fs
brctl                   freshsebool            mm-qjpeg-enc-test      rtccd3
bridgemgrd              fsck.exfat             mm-qomx-ienc-test      setup_fs
btnvtool                ftmdaemon              mm-vdec-omx-test       sfotahelper
ccm_gen_cert            hci_qcomm_init         mm-venc-omx-test720p   StoreKeybox
cplay                   hostapd_cli            mm-video-driver-test   subsystem_ramdump
crda                    hvdcp                  mm-video-encdrv-test   syscheck
curl                    imsdatadaemon          mobicore-presetup.sh   tc
diag_callback_client    ims_rtp_daemon         mobicore-startup.sh    test_diag
diag_dci_sample         ip                     n_smux                 tinycap
diag_klog               irsc_util              odekeymgr              tinymix
diag_mdlog              isdbtmmtest            olsrd                  tinypcminfo
diag_socket_log         jackd                  PktRspTest             tinyplay
drmdiagapp              keymaster_test         port-bridge            tlc_server
dsdnsutil               lpm                    profiler_daemon        tlcWrapperApp
ds_fmc_appd             macloader              qmiproxy               usbhub
e2fsck                  mcStarter              qrngp                  usbhub_init
ebtables                mfgloader              qrngtest               vpnclientpm
epmd                    mkfs.exfat             qseecom_sample_client  wcnss_filter
flatland                mldaemon               qseecom_security_test  wdsdaemon
fmconfig                mm-jpeg-dec-test       radish                 wlandutservice
```

Native Libraries

- Installed to /system/lib/ or app directory
- Allows Java to communicate with C++ via Java Native Interface (JNI)
- **Any application can read these**

Device Driver Interactions

- Usually in /dev/
- Very dangerous if exposed to applications

```
root@android-assessment:/DevTesting/[redacted]system-libs# dtf libinfo libdm-systemaccess.so
Dev grep:
/dev/socket/dmagent
[INFO] Imports socket!
00001fe9 T Java_com_htc_engine_system_SystemAccess_ConnmoDnsSetting
00001d61 T Java_com_htc_engine_system_SystemAccess_CopyFileCtl
00004de5 T Java_com_htc_engine_system_SystemAccess_DMFolderPermissionControl
00004ced T Java_com_htc_engine_system_SystemAccess_DcmoBtdue
00004afd T Java_com_htc_engine_system_SystemAccess_DcmoCameradue
00004909 T Java_com_htc_engine_system_SystemAccess_DcmoCameraenb
00004a01 T Java_com_htc_engine_system_SystemAccess_DcmoGpsdue
00004bf5 T Java_com_htc_engine_system_SystemAccess_DcmoWlandue
00004885 T Java_com_htc_engine_system_SystemAccess_EPSTSwitchcontrol
000028dd T Java_com_htc_engine_system_SystemAccess_ExtAUTHALGORW
00004375 T Java_com_htc_engine_system_SystemAccess_ExtHOMESID1RW
00004231 T Java_com_htc_engine_system_SystemAccess_ExtHOMESID2RW
000045fd T Java_com_htc_engine_system_SystemAccess_ExtMDN1RW
000044b9 T Java_com_htc_engine_system_SystemAccess_ExtMDN2RW
00002145 T Java_com_htc_engine_system_SystemAccess_ExtMDfiveAkeyRW
00004741 T Java_com_htc_engine_system_SystemAccess_ExtMEIDRW
000040ed T Java_com_htc_engine_system_SystemAccess_ExtMIN1RW
00003fa5 T Java_com_htc_engine_system_SystemAccess_ExtMIN2RW
```

OEM Changes & Additions

Using DTF (Device Testing Framework)

“dtf” Basics

- Device testing framework
  - Written in Bash, C, Python (gross)
- “Lead generation”

```
root@android-assessment:/# dtf -h
Android Device Testing Framework (dtf) verision 0.1a
Usage: /repos/dtf/dtf [command] <command_args>
   Core Commmands:
     config       Prints the project's configuration file.
     delprop      Removes a property from the project's configuration.
     getprop      Returns a property from the project's configuration.
     help         Prints this help screen.
     init         Initializes a project.
     local        Display all local modules.
     modules      Print all global and local modules.
     reset        Removes the DTF project from the current directory.
     setprop      Sets or updates a property from the project's configuration.
     shell        Creates a shell on your test device.
     status       Prints metadata about the project.
```

“dtf” Basics

- Project specific configuration file
- Package installer and module support
  - Modules perform all the exciting functionality
  - *dtf <module_name>*

Modules: Data Collection

- Collect files from device:
  - *getsysapps*
  - *getframeworks*
  - *getbins*
  - *getsyslibs*
  - *getpermissions*
- Stores all files locally

Modules: Data Processing

- Application and framework unpacking:
  - *unframework*
  - *unpacksysapps*
- Local database creation:
  - *appdb*
  - *appdexdb*
  - *frameworkdb*
  - *frameworkdexdb*
  - *devdb*
  - *sysservicedb*

Modules: Data Analysis

- 'Diff'ing project against AOSP:
  - *appdiff/appdexdiff*
  - *frameworkdiff/frameworkdexdiff*
  - *sysservicediff*
  - *devdiff*
  - *(provider|service|receiver|activity)diff*
  - *platformdiff*
  - *bindiff*
  - *syslibdiff*

Modules: Data Analysis (cont.)

- Searching for exposure:
  - *readablefiles*
  - *writablefiles*
  - *suidfiles*
  - *nolauncher*
  - *app-metadata*
- CSV of exposed components:
  - *(secretcode|debuggable|activity|service|provider|reciever)csv*

Modules: General Commands

- *libinfo*
  - Searches SO library for JNI calls, sensitive imports, and device interaction
- *secretcode*
  - Sends a SECRET_CODE intent
- *newapp*
  - Creates a new test application (in Smali)
- *classsearch*
  - Searches DEX databases for class name match

Closing Thoughts

- Device OEMs and carriers have **a lot** to learn.
  - 1999 style issues
- Issues are extremely apparent, given the correct tools
- Be careful how much trust you put in your device!

Questions?