Android Forensics
Mobile Forensics World 2009

Presented by Andrew Hoog

May 29, 2009

Android mobile platform

- July 2005: Google acquires Android, Inc.
  - Andy Rubin – now Sr. Director of Mobile Platforms at Google
- Nov 2007: Open Handset Alliance unveiled
  - Originally 34 members, now 47 firms including mobile operators, software companies, chip makers and handset makers
  - Nokia and AT&T are not yet members
- Open source, Apache 2.0 and GPLv2 licenses
  - (source | developer | market).android.com

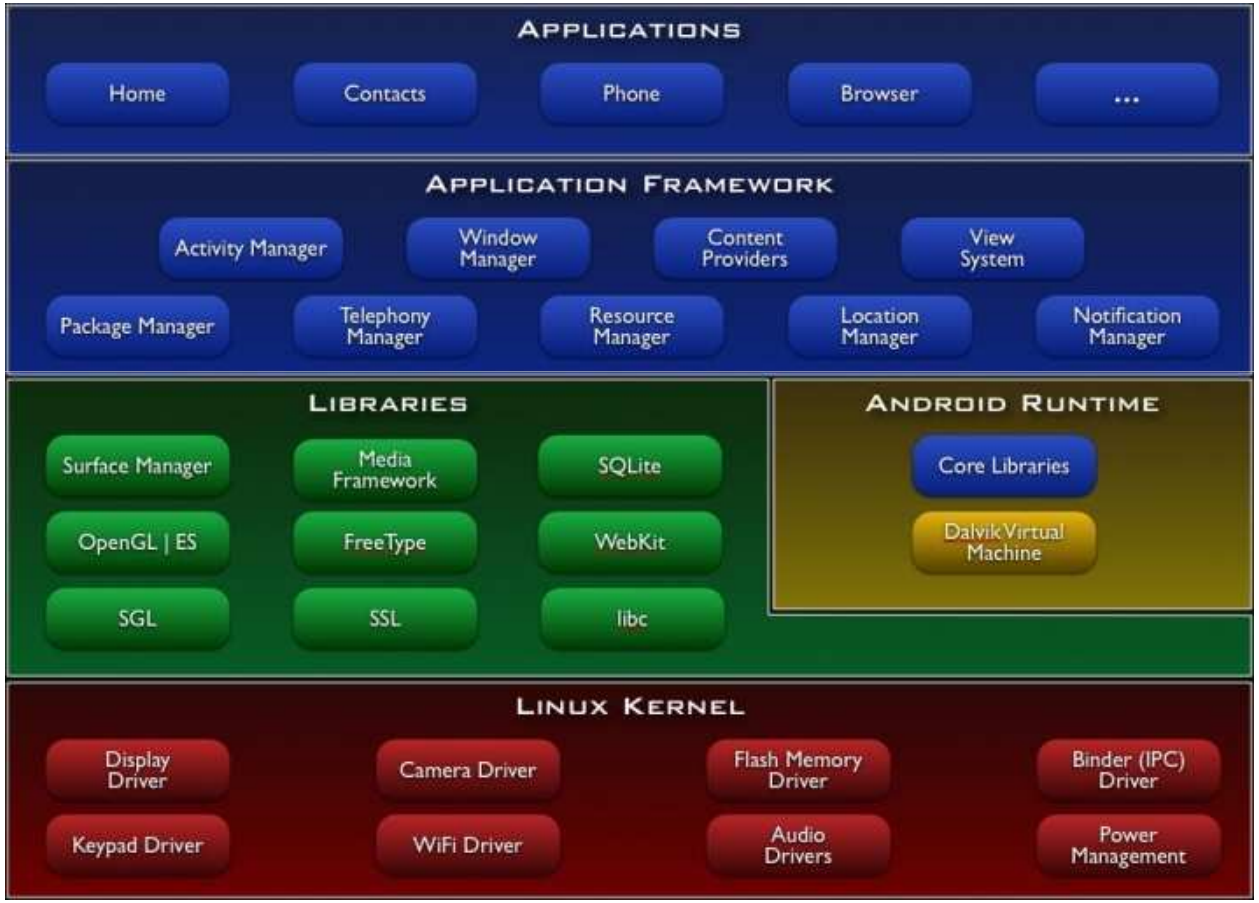
Android devices (18-20 new in 2009)

- 6 currently released
  - T-Mobile G1 and G2 (by HTC), Samsung i7500, Google ADP1 and several others
- 10+ run with Android installed after market, e.g. HTC Hero which was developed for Windows but Android-ported
- Netbooks
  - HP actively researching and testing, Skytone Alpha 680 at $100-$200
- Carriers/handset soon to release devices
  - Verizon, T-Mobile, Vodafone, Motorola, Lenovo, Far EasTone
- Other devices
  - Garmin, Sony Ericsson (DVR), Acer, Huawei, Sharp (large, networked copiers), medical devices

Android technical overview

- Based on Linux 2.6 kernel
- Porting to many processors, including Intel, ARM, MIPS, etc.
- Dalvik virtual machine
- SQLite for structured data storage
- Bionic C library (BSD-derived implementation)

Android architecture



Android SDK

- Application development is done in Java (Dalvik VM)
  - Not standard JVM, JME, etc.
- SDK free for anyone to download and use
  - Contains helpful documentation
  - Integrated emulator (with root access)
- Each application run in separate VM, with separate process and user id.
  - In AndroidManifest .xml describes application and allows data sharing

Android updates

- Responsibility of each carrier
- For G1, US and UK releases
  - RC7, RC8: UK
  - RC19, RC28, RC29, RC30, RC33 and then 1.5 (CRB43)
- Anyone can fork Android code. Can also contribute back if registered. Google accepts changes into main branch.

How to update your G1

- OTA

- Manual (preserves user data)
  - Download signed update, copy to SD Card and rename to update.zip, boot into recovery mode, Alt-L, Alt-S, reboot after complete.

- Flash/Factory reset (wipes user data)
  - Download signed update, unzip and extract DREAIMG.nbh, copy to root of flash, power off, enter bootloader (hold Camera and power), press Power to flash, reboot with Phone + Menu + Power

How to root your G1

- Firmware must be RC29 (RC7 UK) or lower
- If you have newer firmware, you can flash previous firmware but all user data is destroyed (research focused on work around)
- From Home screen, type Enter twice, “telnetd”, enter
- Telnet to localhost (or Wifi IP), you now have #

How to keep root

- At #, you update the recovery.img from SD Card
  - mount -o rw,remount -t yaffs2 /dev/block/mtdblock3 /system
  - cd sdcard
  - flash_image recovery recovery.img
  - cat recovery.img > /system/recovery.img
- Optionally update Hard SPL which allows applying future updates, creating backups, apply source Android build, etc.
- Optionally apply JF updates which shadow official G1 releases by a few days

Android file systems (mount)

- root@wintermute:/scratch/android# adb shell mount
- rootfs on / type rootfs (ro)
  tmpfs on /dev type tmpfs (rw,mode=755)
  devpts on /dev/pts type devpts (rw,mode=600)
  proc on /proc type proc (rw)
  sysfs on /sys type sysfs (rw)
  tmpfs on /sqlite_stmt_journals type tmpfs (rw,size=4096k)
  /dev/block/mtdblock3 on /system type yaffs2 (ro)
  /dev/block/loop0 on /system/modules type cramfs (ro)
  /dev/block/loop1 on /system/xbin type cramfs (ro)
  /dev/block/mtdblock5 on /data type yaffs2 (rw,nosuid,nodev)
  /dev/block/mtdblock4 on /cache type yaffs2 (rw,nosuid,nodev)
  /dev/block/mmcblk0p1 on /sdcard type vfat (rw, dirsync, nosuid, nodev, noexec, uid=1000, gid=1000, fmask=0711, dmask=0700, codepage=cp437, iocharset=iso8859-1, utf8)

Android MTD

- G1 raw flash device, need Flash Translation Layer (FTL)
- Memory Technology Device (MTD) subsystem for memory devices (esp. Flash), provides FTL
- Allows OS to interact with NAND as standard block device
- Special characteristics require different file system approach

Android MTD blocks

- root@wintermute:/scratch/android# adb shell cat /proc/mtd
  - dev: size erasesize name
    mtd0: 00040000 00020000 "misc“
    mtd1: 00500000 00020000 "recovery“
    mtd2: 00280000 00020000 "boot“
    mtd3: 04380000 00020000 "system“
    mtd4: 04380000 00020000 "cache“
    mtd5: 04ac0000 00020000 "userdata“
    mtd6: 10000000 00020000 "msm_nand"

- OS creates /dev/mtd/mtd0 and /dev/mtd/mtd0ro devices
  - Don’t try to image from /dev/block/mtdblock devices

Android YAFFS2

- Yet Another Flash File System 2
- Open source
- Have to compile tools/kernel module yourself (some optional support in newer kernels)
- Provides
  - Wear leveling (good for forensics as data retained on device longer)
  - Much faster and YAFFS and JFFS, uses less RAM
  - Supports many flash geometries
  - Built in error correction (important to use nandread/nandwrite tools!)
- "Silly Old Name" – looked at kernel/fs/yaffs2/yaffs_guts.c

Hex view of mtd6ro.dd, USB info

- Can see start of SPL

```
02400000   0E 00 00 EA  30 2E 39 35  2E 30 30 30  30 00 00 00  ....0.95.0000...
02400010   44 72 65 61  6D 20 53 50  4C 20 45 56  54 00 00 00  Dream SPL EVT...
02400020   53 68 69 70  70 65 64 00  00 00 A0 E1  00 00 A0 E1  Shipped.........
```

- USB shows:

```
[267646.230676] scsi 7:0:0:0: Direct-Access     HTC      Android Phone    0100 PQ:
0 ANSI: 2
[267646.245813] sd 7:0:0:0: [sde] Attached SCSI removable disk
[267646.245943] sd 7:0:0:0: Attached scsi generic sg5 type 0
```

Android forensics acquisition techniques

- Android Debug Bridge
- Nandroid backup
- dd/cat image of NAND
- Proof of concept software app
- Commercial tools
- Theoretical
  - Simulated SD Card to swap known good update.zip after initial read
  - Serial commands over USB

- SD Card

Android forensics post-acquisition techniques

- YAFFS2 tools
- Scalpel/foremost
- Logical file system examination
- FAT32 analysis of SD Card
- Dexdump to disassemble applications (interesting technique for the inevitable spyware applications)

- Many of the same techniques you use today

File system


```
drwxrwx---    1 1000     2001          2048 Sep  3 18:36 cache
drwxrwx--x    1 1000     1000          2048 Oct 24 22:44 data
-rw-r--r--    1 0        0               93 Jan  1  1970 default.prop
drwxr-xr-x   11 0        0             2400 Feb 25 03:08 dev
lrwxrwxrwx    1 0        0               11 Feb 25 03:08 etc -> /system/etc
-rwxr-x---    1 0        0           102464 Jan  1  1970 init
-rwxr-x---    1 0        0             1567 Jan  1  1970 init.goldfish.rc
-rwxr-x---    1 0        0             8780 Jan  1  1970 init.rc
-rwxr-x---    1 0        0             1189 Jan  1  1970 init.trout.rc
dr-xr-xr-x   73 0        0                0 Jan  1  1970 proc
drwx------    2 0        0                0 Jan  1  1970 root
drwxr-x---    2 0        0                0 Jan  1  1970 sbin
d---rwxrwx    2 1000     1000          4096 Feb 25 12:35 sdcard
drwxrwxrwt    2 0        0               40 Feb 25 11:35 sqlite_stmt_journals
drwxr-xr-x   12 0        0                0 Jan  1  1970 sys
drwxr-xr-x    1 0        0             2048 Feb 24 22:07 system
```

Interesting files/directories

- /data/
  - dalvik-cache: .dex files that were run
  - anr: debug/thread info with timestamps
  - app: .apk files (install bundle for applications)
  - data: subdirectories per application with sqlite databases
  - misc: dhcp, wifi, etc. files
  - system:
    - packages.xml (installed applications)
    - checkin.db (lot of connection up/down info)
    - etc.

Android Debug Bridge

- A tool that allows interaction with an Android device over USB
  - Runs on workstation as a client/daemon
  - Talk to Android adbd daemon
  - Daemon runs as root on emulator/root'd phone, otherwise very limited privileges
- Can send shell commands (dd, ls, mount, cat, ps, date, uptime, uname -a, mount, etc.)
- Can recursively push/pull files (logical)
  - adp pull|push <src> <dest>
  - I had to run as root on forensic workstation

ADB data pull


```
root@wintermute:/home/ahoog/adb-pul# adb pull /data data/
pull: building file list...
<snip>
pull: /data/miscrild_nitz_long_name_31026 -> data/misc/rild_nitz_long_name_31026
pull: /data/misc/akmd_set.txt -> data/misc/akmd_set.txt

712 files pulled. 0 files skipped.
963 KB/s (208943249 bytes in 211.671s)
```

I was able to pull 1,255 files (19MB) in about 90 seconds.

Nandroid backup

- Fully preserve file system and data
- Preserves configuration settings
- Must run on device with root access

- svn co http://svn.infernix.net/nandroid/

Nandroid output (to SD Card)


```
root@wintermute:/home/ahoog/android-root/nandroid/nandroid# ./nandroid.sh g1_nandroid
nandroid v2.1
mounting system and data read-only on device
start adb portforward on port 4531
checking free space on cache
pushing tools to /cache: dump_image-arm-uclibc... done
Dumping splash1 to g1_nandroid/splash1.img... done, verifying...OK
Dumping splash2 to g1_nandroid/splash2.img... done, verifying... OK
Dumping boot to g1_nandroid/boot.img... done, verifying... OK
Dumping recovery to g1_nandroid/recovery.img... done, verifying... OK
Dumping misc to g1_nandroid/misc.img... done, verifying... OK
Dumping system to g1_nandroid/system.tar... done, verifying... OK
note: fakeroot found but /home/ahoog/android-root/nandroid/nandroid/mkyaffs2image-x86_64 is statically linked
replace with a dynamically linked copy to enable fakeroot support
Extracting system.tar to g1_nandroid/HT849GZ14163-system-tmp... runnig mkyaffs2image...done
Dumping data to g1_nandroid/data.tar... done, verifying... OK
note: fakeroot found but /home/ahoog/android-root/nandroid/nandroid/mkyaffs2image-x86_64 is statically linked
replace with a dynamically linked copy to enable fakeroot support
Extracting data.tar to g1_nandroid/HT849GZ14163-data-tmp... runnig mkyaffs2image...done
Dumping cache to g1_nandroid/cache.tar... done, verifying... OK
note: fakeroot found but /home/ahoog/android-root/nandroid/nandroid/mkyaffs2image-x86_64 is statically linked
replace with a dynamically linked copy to enable fakeroot support
Extracting cache.tar to g1_nandroid/HT849GZ14163-cache-tmp... runnig mkyaffs2image...done
removing tools from /cache: dump_image-arm-uclibc... done
unmounting system and data on device
generating md5sum file...done
Backup successful.
```

Using dd/cat to acquire image

- root@wintermute:/scratch/android# time adb shell dd if=/dev/mtd/mtd6ro of=/sdcard/mtd6ro.dd bs=4096
- 65536+0 records in
  65536+0 records out
- real 2m14.849s
  user 0m0.004s
  sys 0m0.008s
- Can also use cat

Android forensics using application development

- Android has enforced security at the application level very well
- Framework provides for applications sharing data
  - i.e. Twitter applications need access to SMS data. Default install of my important applications (contacts, call logs, SMS, etc.) allow information sharing, if the user approves
- Commissioned an Java developer to write a proof of concept application which will
  - Read data from aforementioned applications
  - Write to CSV on SD Card
  - Will provide as part of our book, can be easily extended/improved

Commercial support for Android Forensics

- Known vendors who support (or plan to support) Android
  - Cellebrite
  - XRY
  - Paraben
  - Others? Please speak up

- Like any situation, forensic analysis should test the tools, understand how they work and be able to explain if needed.

Serial over USB (theoretical)

- HTC Dream service manual mentioned Serial/USB connection
- Cabling was reverse engineered, directions at:
  - http://www.instructables.com/id/Android_G1_Serial_Cable/
- Requires experimentation (or more service manuals in the wild)
  - Using techniques such as USB Snooping, establish protocol and debug communication
  - Attempt to reconstruct available commands

Simulated SD card (theoretical)

- When G1 runs a signed update it:
  - Reads update.zip, verifies RSA signature
  - Re-reads update.zip (no check this time) and applies update
- Simulated SD Card would swap update.zip with new update after first read
- New update.zip would make the update process non-destructive, allow tools/techniques for acquiring image of data files

Android Forensic Resources

- http://viaforensics.com/android

- 113 page HTC Dream service manual
- This presentation
- Updates on the Android Forensics book
- Discussion boards

- We need more research…email if interested