Towards a General Collection Methodology for Android Devices

Timothy Vidas
ECE/CyLab

Chengye Zhang
INI/CyLab

Nicolas Christin
INI/CyLab

Outline

- Motivation
- Related Work
- Android Background
- Collection Process
- Discussion & Future work
- Conclusion

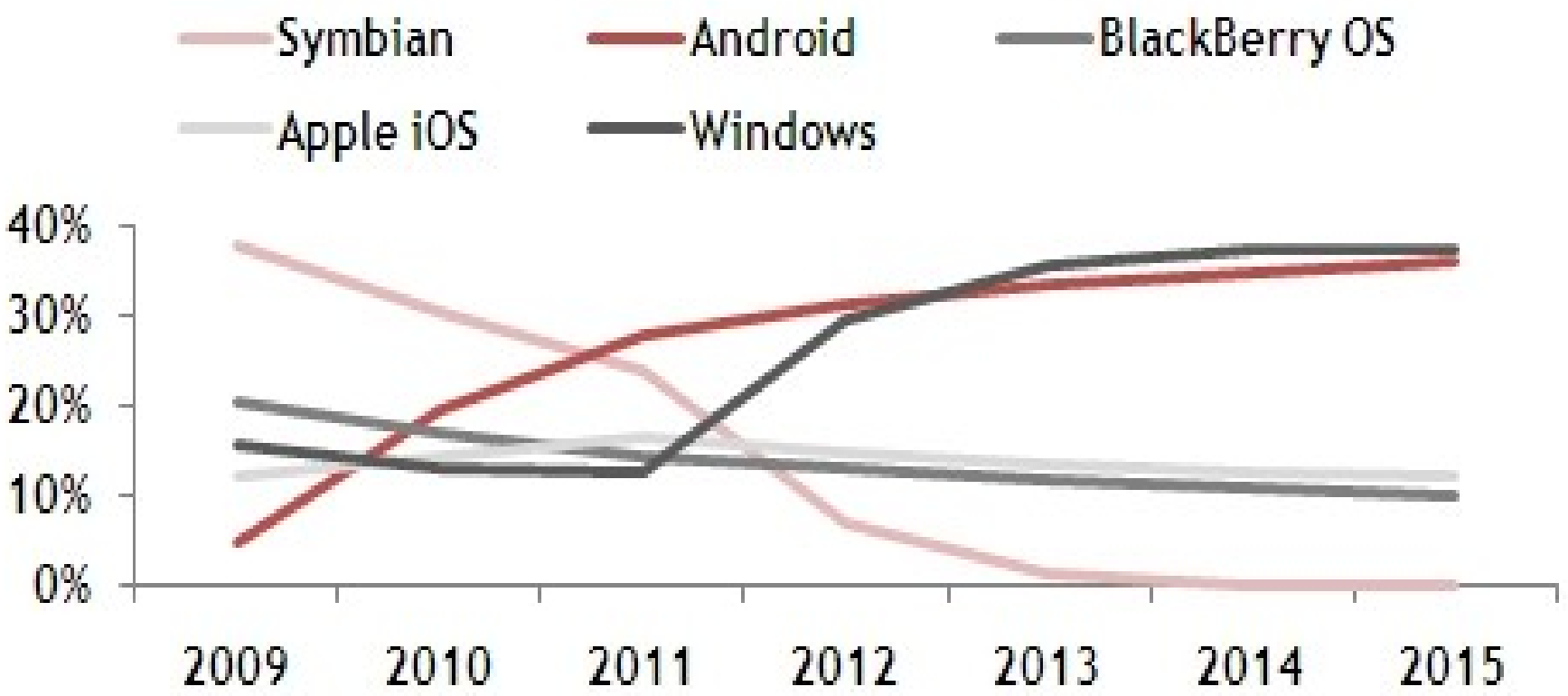
Motivation

Fragmentation among mobile devices makes it more difficult to collect data

Devices have different:

- Form factors
- Operating systems
- Memory layouts
- Connectors

Android Market Share



“There are now over 500,000 Android devices activated every day, and it's growing at 4.4% w/w”

~Andy Rubin, Google

Mobile Phones

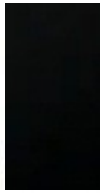
a big

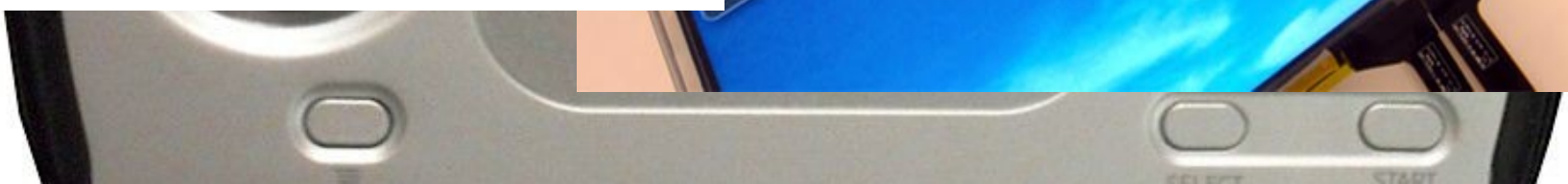
Tablets


MOTOROLA
Google
Music
Video
Market
Talk
Maps
Browser
ASUS
Taipei
Partly Sunny
25 °C
3mails
Mar 22
ThinkPad
lenovo


下午 5:32
Google
飞信
浏览器
淘宝
新浪微博
机器人塔防
QQ
QQ农场
视频
GTalk
谷歌地图
媒体库
电子书
PANDIGITAL
Bookstore


23:39
No SIM card!
Sony Ericsson
Friday





Android's Ubiquity

Android's Ubiquity brings unprecedented software commonality to this wide range of devices.

Related

Mobile Phone forensics is not new.

Many vendors have already or are bringing collection products to market

Techniques borrowed from the phone hacking or phone modding community

To Root or not to Root

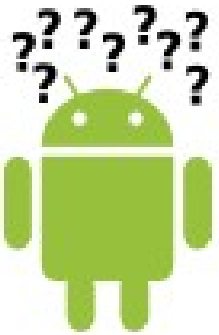
"Phone modding" definition:
somehow obtaining elevated privileges on a phone in it's normal operating mode.

Many methods of rooting exist, rooting is quite popular among tech savvy and non technical people alike

To Root or not to Root

Rooting may not be desirable for Digital Forensics, for example

- Often device and software version specific
- Data in user data stores may be modified
- A rooted device may be left in a lesser security state



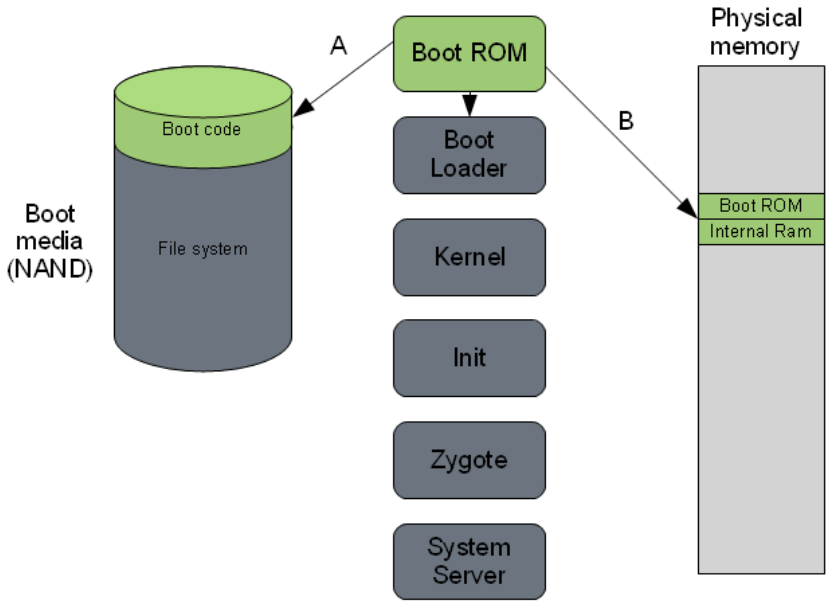
Before getting into the technique, it's worth looking at the android boot process a bit...

Boot Process

Many parts of the Android boot process are akin to that of a typical Linux system.

A) The Boot ROM (software on the board) locates boot media.

B) Boot code is loaded from media into memory. Execution is transferred to the boot code.



Graphic: elinux.org

Boot Process

Like many boot loaders (think GRUB on a typical Linux machine), the loader here is staged. Stage1 and Stage2.

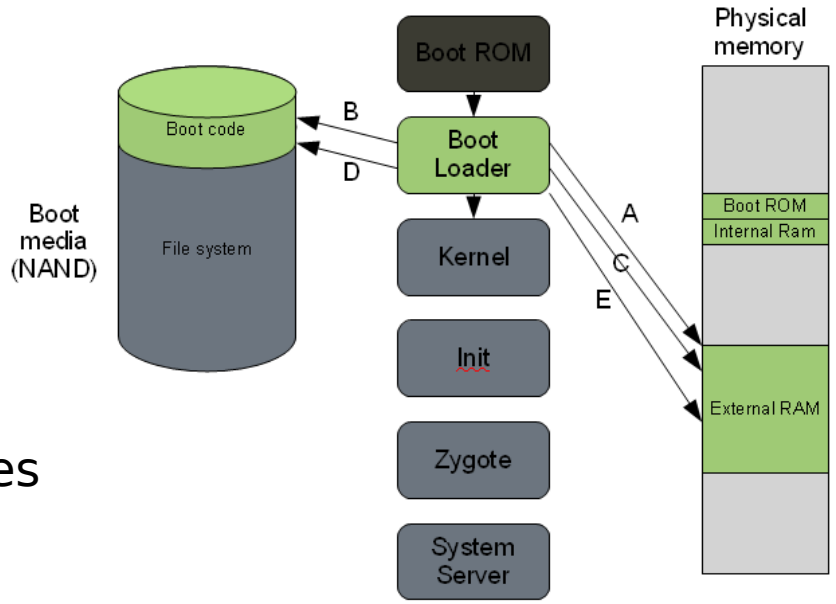
A) Stage1 will setup RAM

B) Stage1 will load Stage2 into RAM

C) Stage2 is executed (loads other binary images such as “modem” for use by the phone hardware, initializes hardware, etc)

D) Stage2 loads the Linux kernel from boot media, decompresses, configures options,

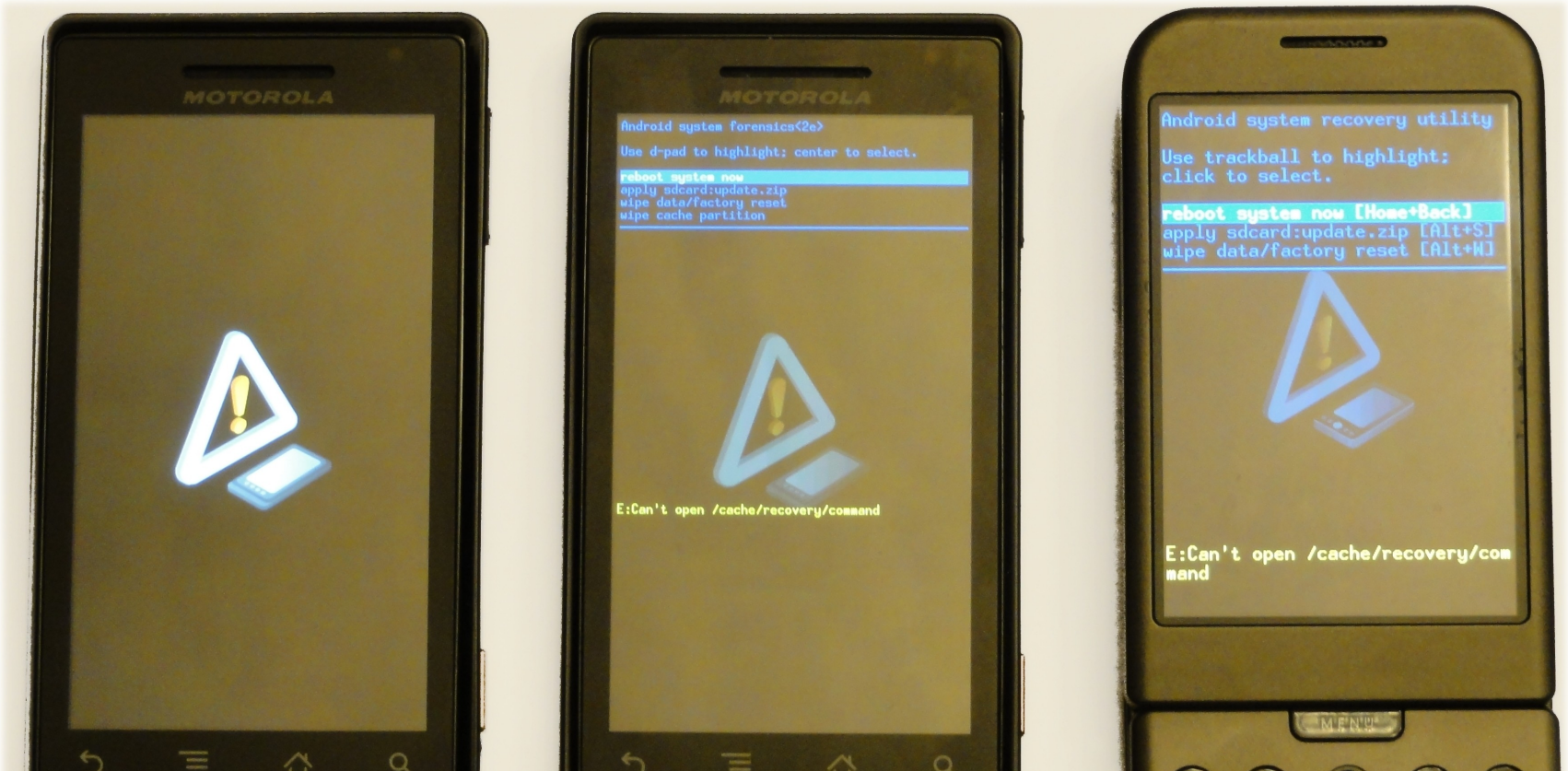
E) transfers execution to the kernel



Graphic: elinux.org

Recovery Mode

Recovery mode is a standard boot mode invoked by holding keys during the Stage1 bootloader altering the typical boot process.



Recovery Mode

Instead of loading the typical boot image, execution is diverted to a special recovery boot image

Path	Name	FS	Mount Pt	Description
/dev/mtd/mtd0	pds	yaffs2	/config	Config data
/dev/mtd/mtd1	misc			Memory Partitioning Data
/dev/mtd/mtd2	boot	bootimg		Typical boot image
/dev/mtd/mtd3	recovery	bootimg		Recovery mode boot image
/dev/mtd/mtd4	system	yaffs2	/system	System files, System apps, etc
/dev/mtd/mtd5	cache	yaffs2	/cache	Cache files
/dev/mtd/mtd6	userdata	yaffs2	/data	User Data (apps, settings, etc)
/dev/mtd/mtd7	kpanic			Crash log

Recovery Collection

A typical recovery image allows for applying updates, wiping user data, etc

No user data is stored in the recovery partition

The contents of the recovery image has no effect on the typical boot and operation of the device

No Radio

Recovery Collection

A custom bootimg is crafted that extends the functionality of the a typical bootimg

Binaries are added:
adbd, su, nanddump

Files are edited:
default.prop, init.rc

(enable adb, start the daemon, etc)

Filesystem permissions are adjusted to facilitate the above

Recovery Collection

Executing in this context gives full access to data on the phone

Using nanddump permits the collection of OOB (eg spare) data that would likely otherwise be lost.

Recovery Collection

Technique was used for DFRWS2011 challenge! :-D

Discussion

- This technique:
  - ✔ Applies to a wide range of devices
  - ✔ Does not modify any storage areas that hold user data
  - ✔ Permits a priori setup
  - ✗ Doesn't take volatile data into account
  - ✗ Requires different flashing tools based on manufacturer

Moving forward

- Extended functionality/usability
- More devices
- “Unified bootimg”
- Comprehensive list of boot modes (and behaviors)

Conclusion

- Recovery boot method takes advantage of commonality across Android devices
- Comprehensively collects data with no impact to data areas used under normal operation

Thanks!

Tim Vidas
tvidas@cmu.edu