



# Towards a General Collection Methodology for Android Devices

Timothy Vidas, Chengye Zhang, Nicolas Christin

Presented by Muhammad Omer Qureshi.

# Organization:

- Introduction.
- Motivation.
- Mobile Forensics: Past & Present.
- Android: Why Android ?
- Android: Ubiquitous Nature.
- Android Boot mode.
- Android Boot Process.
- Filesystem and Partition.
- Recovery Images.
- Collection Process.
- Forensically Important locations.



# Introduction.

This paper present a general collection methodology for Android based devices, by exploiting commonalities provided by Android system and using recovery mode.

# Motivation

Fragmentation among mobile devices makes it more difficult to collect data.


Devices have different:

- Form factors
- Operating systems
- Memory layouts
- Connectors



# Mobile Forensics: Past & Present

## Past:

- Forensic analysis of SIM cards. 
  - Comprises of Processor & memory.
  - Store many files with unique data for e.g SMS, Recently dial numbers, MSISDN.
  - SIM card readers are used to copy contents to external device.
  - It also store location information.
  - Anti-Forensics techniques can be used to destroy evidences.

# SIM Card Files:

Phase	Phase ID	1 byte
SST	SIM Service table	5 bytes
ICCID	Serial Number	10 bytes
LP	Preferred languages	variable
SPN	Service Provider name	17 bytes
MSISDN	Subscriber phone number	variable
AND	Short Dial Number	variable
FDN	Fixed Numbers	variable
LND	Last Dialed numbers	variable
EXT1	Dialling Extension 1	variable
EXT2	Dialling Extension 2	variable
GID1	Groups 1	variable
GID2	Groups 2	variable
SMS	Text Messages	n * 176 bytes
SMSP	Text Message parameters	variable
SMSS	Text message status	variable
CBMI	Preferred network messages	variable
PUCT	Charges per unit	5 bytes
ACM	Charge counter	3 bytes
ACMmax	Charge limit	3 bytes
HPLMNSP	HPLMN search period	variable
PLMNsel	PLMN selector	variable
FPLMN	Forbidden PLMNs	12 bytes
CCP	Capability configuration parameter	14 bytes

# Attack on SIM Cards:

- Removal of evidence.
- Changing of Information, this can results in
  - Impersonation.
    - This could result in identity theft.
    - Need to find  $K_i$ , the encryption key. It is only possible if the algorithm contain weakness.
    - Need to produce new SIM card.
    - Conclusion: Required Technical skills and tools, not a easy task.

# Past:

- Forensic analysis of Not-So-Smartphones.
  - IMEI
  - Stored Computer Files
  - Logged incoming calls and dialed numbers
  - Stored Calendar Events
  - Stored Audio Recordings
  - Short Dial Numbers
  - Text Messages



# Attack on Phones:

- Phone software:
  - Can be flashed to remove constraint but can be used to modify the contents in the memory of the phone.
- Change of IMEI:
  - It is necessary to change IMEI number of stolen phone.
  - The stolen phone IMEI number is blacklisted in EIR of GSM network.

# Past:

- Electronic Information.
  - The network provider can provide following details.
    - Customer name and address.
    - Billing account details.
    - SIM serial number.
    - Call Data Records.
    - Location Information.

# Attacks on Network

- GSM was believed to be immune from security breaches.
  - Recent studies and experiments shows that transmission protocol has flaws due to
    - Lack mutual authentication
    - Lack of mandatory encryption
    - Network can order the MS to turnoff or turn on encryption.

# Present:

- Standard Operating system.
  - Android.
  - iOS.
  - Blackery OS.
  - Windows Mobile.
  - Symbian.
- Standard USB port interface (MOSTLY).
- Challenges
  - Memory Size increase tremendously.
  - Anti-forensics techniques.

# Why Android?

- Huge Market share.
- Open Source.
- Ubiquitous.



# Android: Ubiquitous Nature.

- Smart phone are not only Android based devices, now Android can be found in,
  - Television.
  - Cameras.
  - Tablets.
  - Cars.
  - Media Player

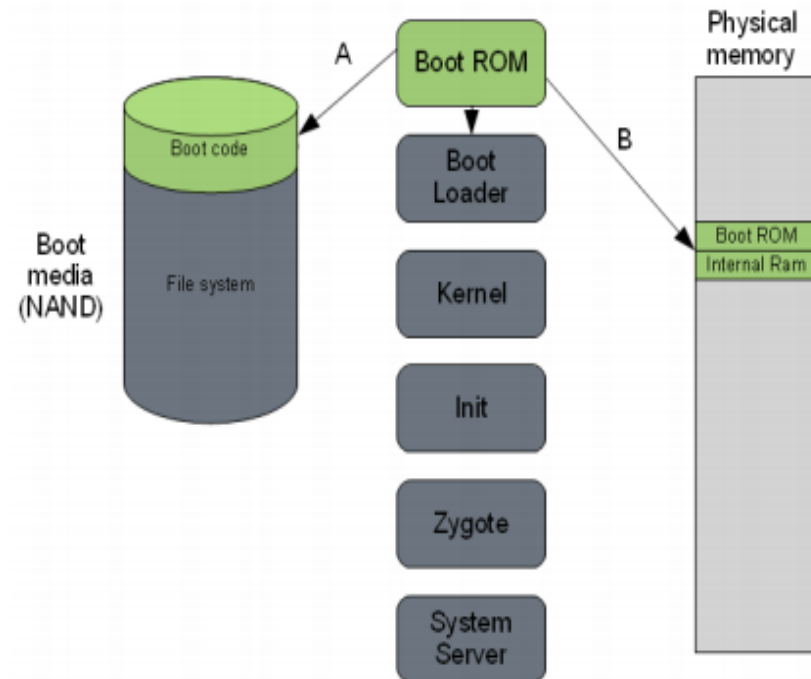


Camera. Reborn.

Samsung  
GALAXY Camera

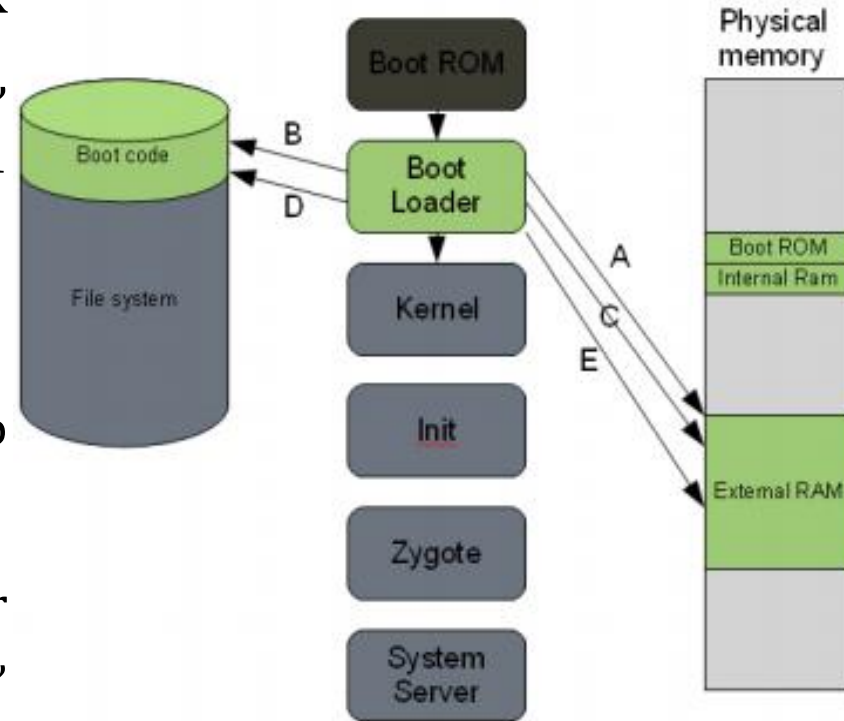
# Android Boot Process.

- 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.



# Boot Process Continue

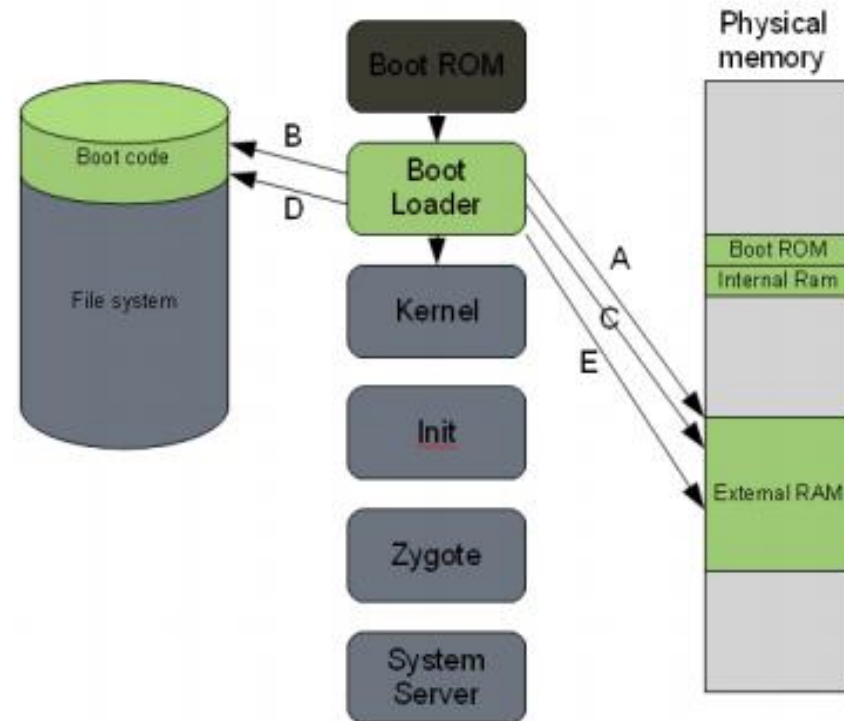
- Like many boot loaders (think GRUB on a typical Linux machine), the loader here is staged. Stage1 & Stage2.
- A) Stage1 will setup RAM
- B) Stage1 will load Stage2 into RAM
- C) Stage2 is executed (loads other binary images such as “modem” for use by the phone hardware, initializes hardware, etc)





# Boot Process Continue

- D) Stage2 loads the Linux kernel from boot media, decompresses, configures options,
- E) transfers execution to the kernel



# Android Boot modes.

- Fastboot Mode:
  - It is protocol used to update the flash file system in Android devices from a host over USB. It allows flashing of unsigned partition images.
- Recovery Mode:
  - By booting a device into “recovery mode,” the normal boot process is circumvented and the boot target is the boot image currently loaded in the recovery partition.

# File system and Partitions.

## Typical Android Portioning:

Path	Name	File System	Mount point	Description
/dev/mtd/mtd0	pds	yaffs2	/config	Configuration data
/dev/mtd/mtd1	misc	—	N/A	Memory Partitioning data
/dev/mtd/mtd2	boot	bootimg	N/A	Bootable (typical boot)
/dev/mtd/mtd3	recovery	bootimg	N/A	Bootable (recovery mode)
/dev/mtd/mtd4	system	yaffs2	/system	System files, Applications, Vendor additions, Read-Only,
/dev/mtd/mtd5	cache	yaffs2	/cache	Cache Files
/dev/mtd/mtd6	user data	yaffs2	/data	User data (Applications)
/dev/mtd/mtd7	kpanic	—	N/A	Crash Log

# File System:

- YAFFS: Yet Another Flash File System.
  - It is highly portable and has been used in different products and application, with multiple different operating system.
  - The memory in NAND flash is arranged in pages but YAFFS arrange memory in chunks.
  - Each chunk can be mapped to one page or multiple pages.
  - 32 – 128 chunks make a block, which can be read or write.
- Newer devices may use EXT4 file system.

# Boot Image.

- An Android boot image is consist of the following,
  - Header.
    - Defined in bootimg.h
    - Magic number ANDROID!
    - Have meta-information about size and memory location.
  - Kernel
  - Ram disk (initrd).
    - Compressed gzip or lzma cpio file.
    - Can be modified to add additional binaries.

# Recovery Image:

- A recovery image is created by modifying

## Default.prop File:

- Ro.secure: It is set to zero in rooted devices.
- ro.debuggable: Enable USB debugging mode, which allows control from PC.

## INIT.RC File:

- It is used to enable adbd (Android Bridge Deamon) service.
- It is done by setting property:persist.services.adb.enable=1

```
#  
# ADDITIONAL_DEFAULT_PROPERTIES  
#  
ro.opengles.version=131072  
persist.sys.usb.config=mtp  
ro.secure=1  
ro.allow.mock.location=0  
ro.debuggable=0  
#  
# VERSION_PROPERTIES  
#  
ro.semc.version.sw=1241-4324  
ro.semc.version.sw_revision=4.1.B.0.431  
ro.semc.version.sw_variant=GENERIC  
ro.semc.version.sw_type=user
```

Default.prop file of Sony Xperia Arc

# Collection Objective:

- Data Preservation:

- The collected data should be exact copy.

- Atomic Collection:

- Data should be collected as atomically as possible with respect to the device being collected. One possible technique is to copy the data when no other operation is performing on the data.

Correctness:

Software must truly copy the data from source to destination and integrity must be preserved in transit.

Determinism:

Repeated collections on the same device, in the same state, should ideally produce identical results.

Usability:

The process must be usable, and occur in a feasible amount of time.

# Collection Process

- The collection process is multi step process.

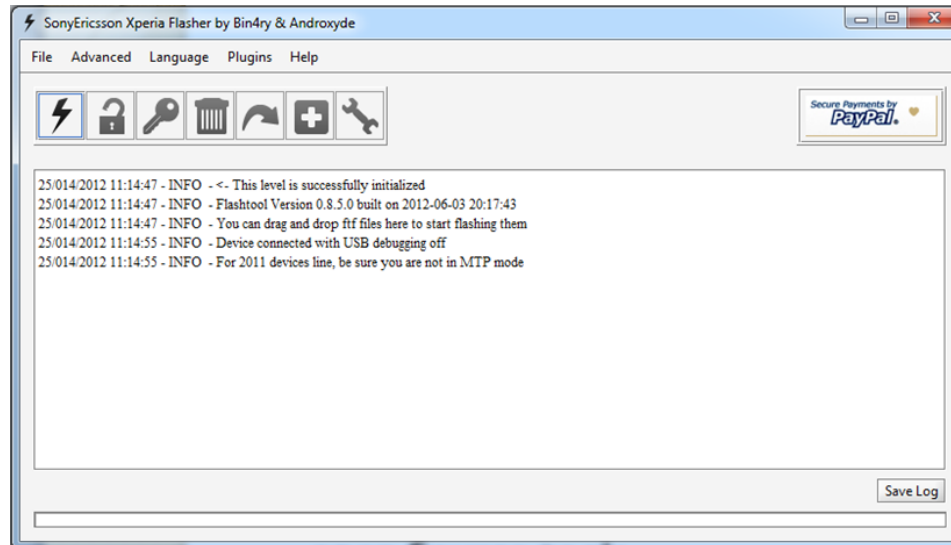




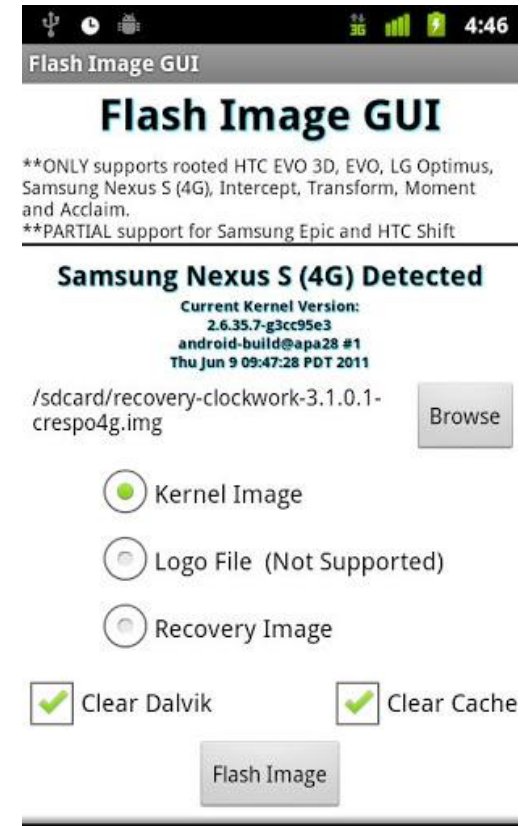
# Flashing:

- There are two way a recovery partition can be flashed.
  - In Device: A recovery image can be flashed to smartphone while operating the device, for e.g. FlashImageGUI.
  - Out of Device: Smartphone needs to be boot in fastboot mode and recovery image can be flashed.
- Flashing of recovery image is device specific.
- It can be done through OEM provided tools or open Source tools.
- Device Specific methods will be discussed in coming slides.

# Flashing tools



Flash tool for Sony Xperia



Flash Image GUI.

# Recovery Mode

- Recovery mode is special boot mode which allow users to by pass normal boot process and give option to
  - Update new ROM.
  - Backup & Recovery Existing ROM.
- Procedure to activate Recovery is different for each device, for e.g. Powering on device while holding volume keys

# Data Collection

- Data need to be collected with meeting the objectives mentioned above.
- The data collection methods used in this paper are
  - NAND Dump.
  - DD

# Motorola Droid

- Form Factor:
  - Touch Screen with Full qwerty keyboard and internal MicroSD card.
- Recovery Mode: Power button + X
  - This boot mode allow the flashing of recovery image.
  - Motorola RSDlite software is used to flash the recovery image.
  - Doesn't accept .img file, a sbf file containing bootimage.img needs to be created.

# Motorola Droid



Flash Mode Motorola Droid booted to flash mode.

# HTC GI

- Form Factor:
  - Touch Screen + Full qwerty keyboard and external MicroSD card.
  - It used ExtUSB port instead of micro USB port.
- Recovery Mode: Power + Home button.
- Flashing Method:
  - Device need to connected in fastboot mode to flash a recovery image.
- Fastboot also allows directly booting to a kernel and ram disk located on a connected computer.

# HTC G1



· Recovery Mode HTC G1 booted to a typical recovery image.



· Fastboot Mode HTC G1 booted to fastboot mode.



# Samsung Captivate

- Form Factor: Touch Screen and 16 GB Internal Storage.
- Recovery Mode: Power + Volup + Voldn
- Flashing Method:
  - Device need to be connected in the flash mode aka download mode.
  - Open Source tool named Odin can be used.
- Samsung Captivate use Robust FAT File System, RFS and OneNAND memory.
- Kernel is required to load module to support RFS.

# Samsung Captivate



- Recovery Mode Samsung Captivate booted to a typical recovery image.

- Partition information typical of a Samsung device.			
Device	Name	Mount point	Description
bml1	boot	-	Primary boot loader
bml2	pit	-	Partition map data
bml3	efs	/efs	Unknown.
bml4	SBL	-	Secondary boot loader
bml5	download	-	Download Mode
bml6 <sup>7</sup>	param	/mnt/lfs	Unknown (lfs)
bml7	kernel	N/A	kernel + initramfs
bml8	recovery	N/A	kernel + initramfs
bml9 <sup>7</sup>	system	/system	Typical/system data (RFS)
bml10 <sup>7</sup>	dbdata	/dbdata	dbcache (RFS)
bml11 <sup>7</sup>	cache	/cache	cache (RFS)
bml12	modem	-	Modem software

# Common Mobile Phone Data Locations

- Phone Contact & Call logs:
  - `/data/data/com.android.providers.contacts/databases/contacts2.db`
- Calendar information:
  - `/data/data/com.android.providers.calendar/databases/calendar.db`
- SMS and MMS messages:
  - `/data/data/com.android.providers.telephony/databases/mmssms.db`
- Gmail and gtalk data:
  - `/data/data/com.google.android.providers.gmail/databases/mailstore.cmu.  
android.<GMAILADDRESS>.db`
- These databases are SQL Lite 3 Databases.