New Acquisition Method Based On Firmware Update Protocols For Android Smartphones

By

Seung Jei Yang, Jung Ho Choi,
Ki Bom Kim and Tae Joo Chang

Presented At

The Digital Forensic Research Conference

DFRWS 2015 USA Philadelphia, PA (Aug 9th - 13th)

DFRWS 2015

New acquisition method based on firmware update protocols for Android smartphones

Seung Jei Yang, Jung Ho Choi, Ki Bom Kim, and Taejoo Chang

2015. 8. 11.

Seung Jei Yang

sjyang@nsr.re.kr

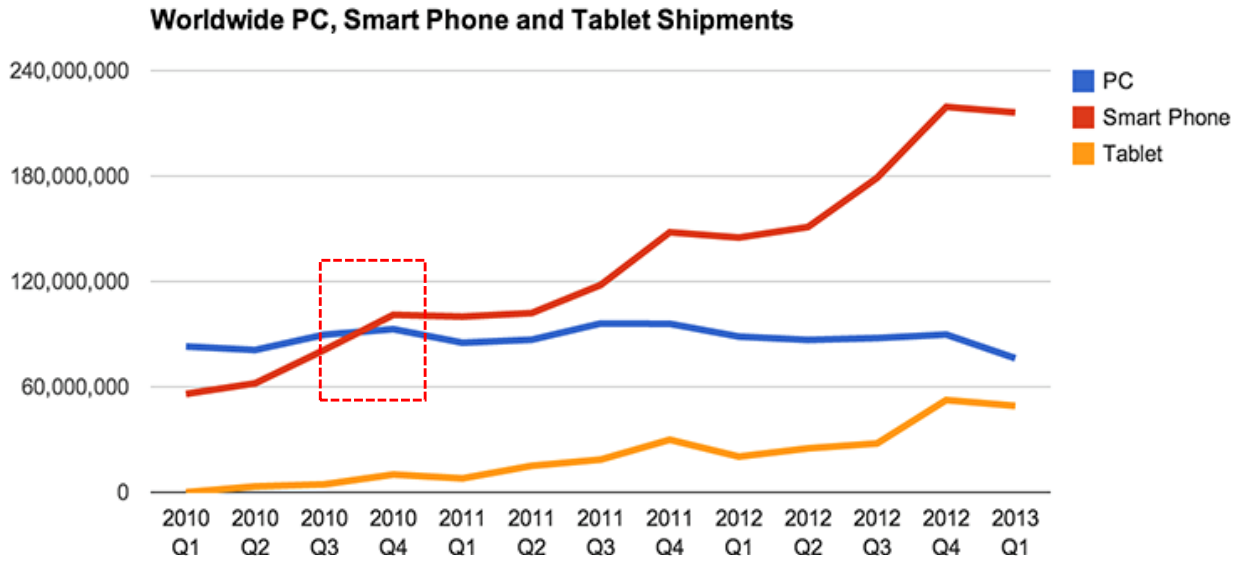
The Affiliated Institute of ETRI

Contents

INTRODUCTION

Explosive growth of smart devices

Smart phones overtake PCs

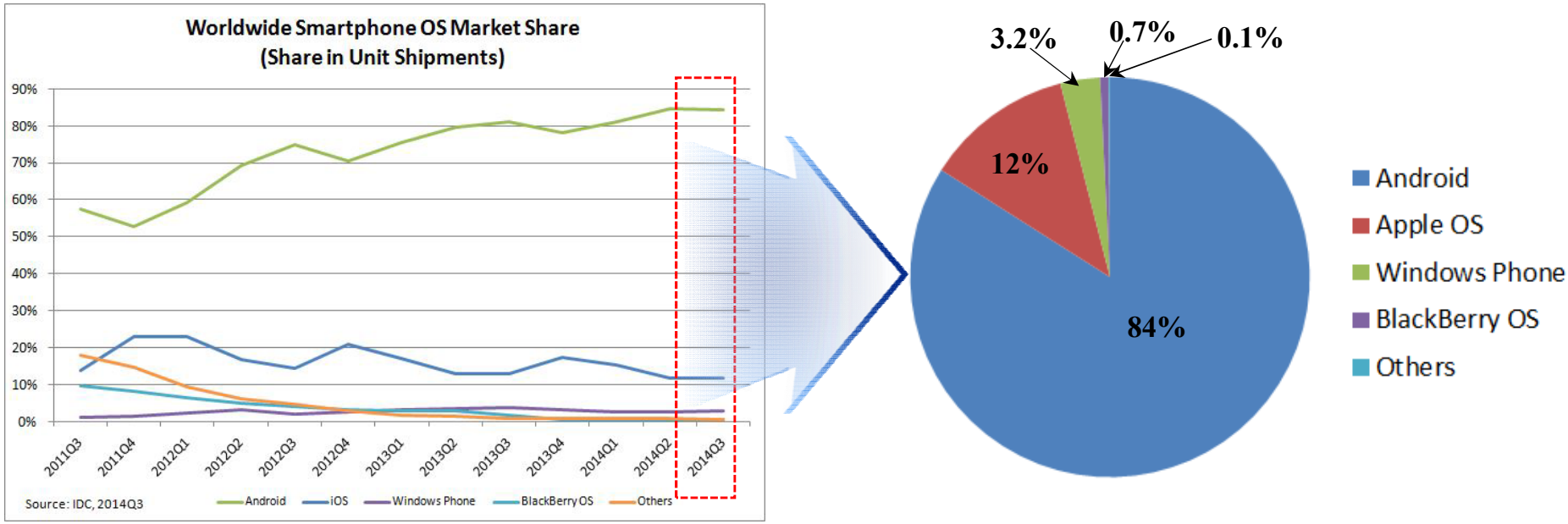


IDC

Smart Revolution !!

Android Forensics

- Increase the importance of Android forensics
  - Approximately 84% of the market share(Q3 2014)



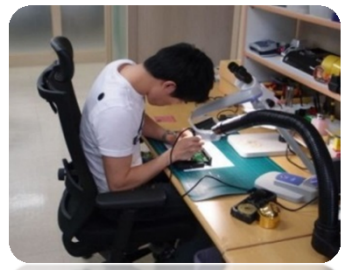
  - Various technologies are emerging continuously not only for personal use but also for business(BYOD)

Physical acquisition of flash memory is particularly necessary

RELATED WORK

1. Software(S/W)-based acquisition

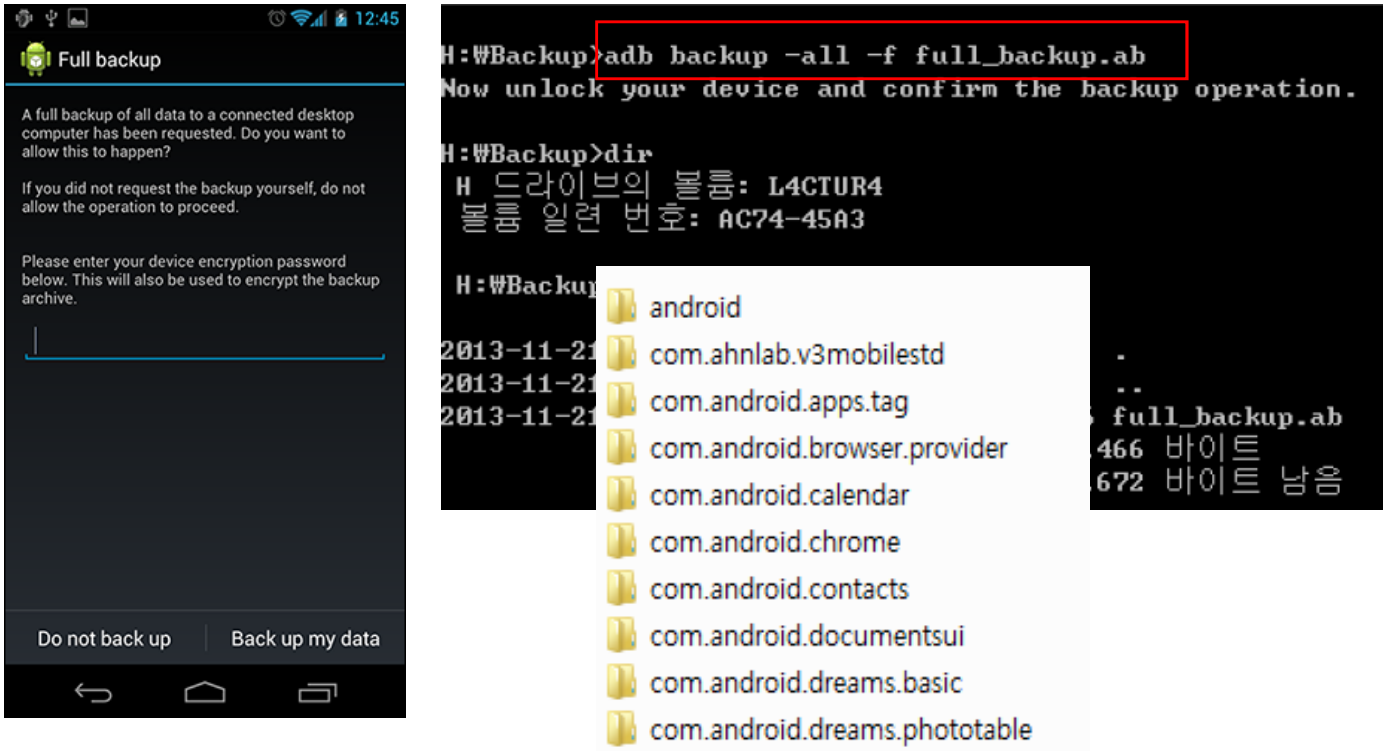
2. Hardware(H/W)-based acquisition



S/W-based acquisition methods(1)

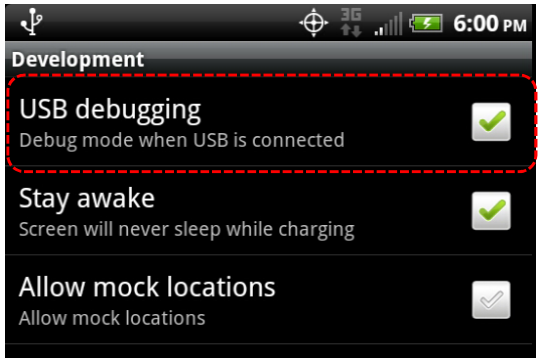
- Logical acquisition methods
  - Acquire stored files from a smartphone after connecting a USB cable
    - Use Content provider, ADB Backup protocol
    - Acquire files such as the call history, pictures, phone books
    - Cannot recover deleted files

ADB Backup



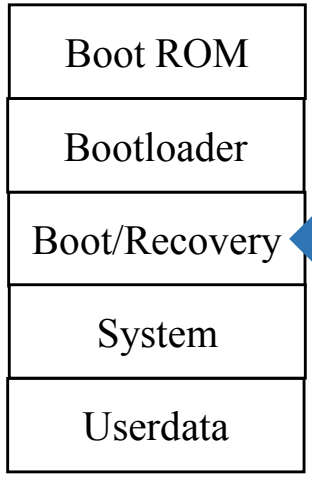
S/W-based acquisition methods(2)

- Physical acquisition methods
  - Extract the overall data directly from flash memory
    - Use the rooting exploitation, the custom recovery image
    - Can recover deleted files
    - Root vulnerabilities are patched whenever new OS is upgraded
    - The integrity of the dumped flash memory is damaged
    - Acquisition is unable when smart phones are screen-locked
    - Apply security technologies (Secure boot, Samsung KNOX)

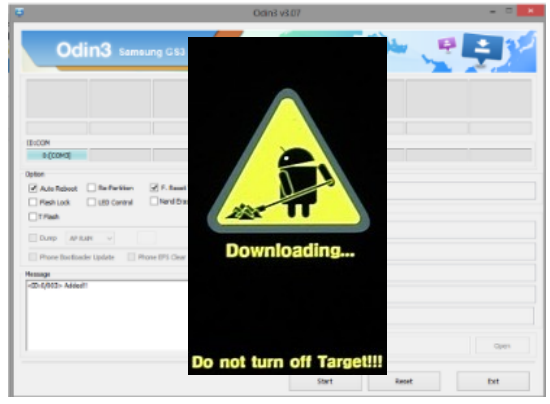



```
C:\adb>adb shell
shell@android:/ $ ls -l /system/xbin
ls -l /system/xbin
-rwxr-xr-x root     shell        59760 2013-09-22 07:23 dexdump
-rwsr-sr-x root     root         91980 2008-08-01 12:00 su
shell@android:/ $ su
su
shell@android:/ #
```

Rooting and use ADB commands



Flashing a custom image and acquisition (DFRWS 2011, 2013)



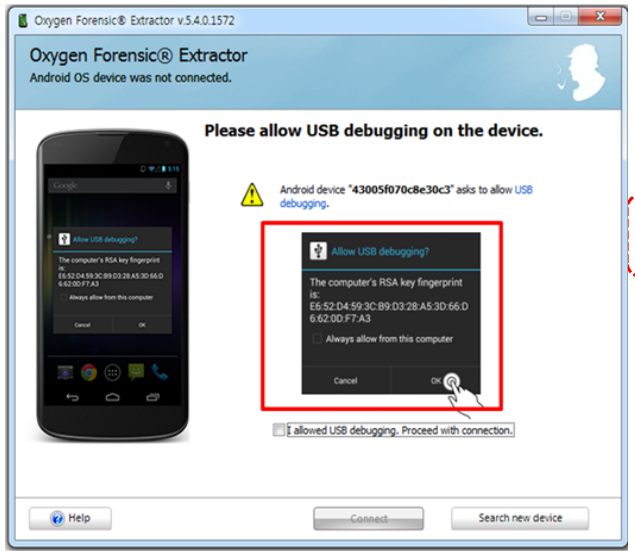
S/W-based acquisition methods(3)

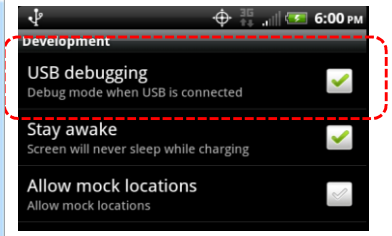
- Commercial forensic tools
  - Cellebrite UFED 4PC, Oxygen Forensics, AccessData MPE+, MSAB XRY
    - Usually use rooting based acquisition method

Cellebrite 4PC

Some Samsung models support physical dumping using a custom bootloader

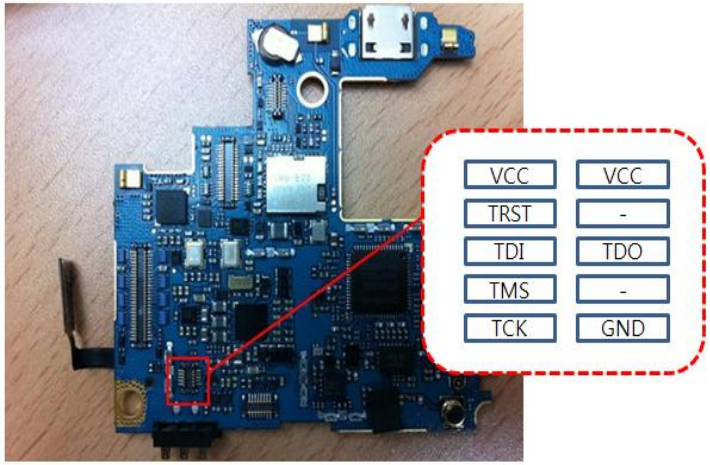
Oxygen Forensics



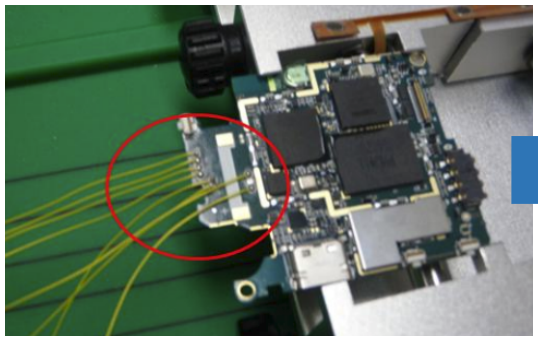


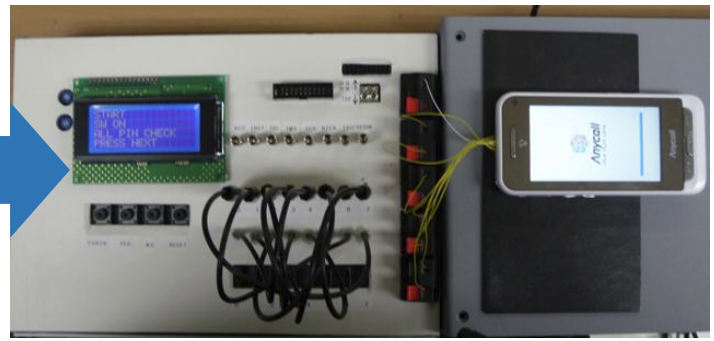
H/W-based acquisition methods(1)

- JTAG-based acquisition method
  - Dump flash memory using the JTAG debug interface
    - Can acquire all flash memory data
    - Have to know the pin map of JTAG on PCB
    - All smart phones don't have JTAG debug interface
    - So long time to acquire data
  - Commercial tools (Flasher tools)
    - RIFF BOX, ORT tool, Z3X box
    - Usually use to fix the bricked phones
      - These tools are not considered as general forensic tools



JTAG-based acquisition method

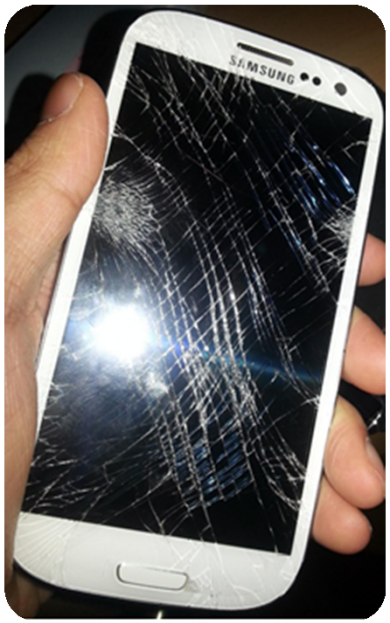


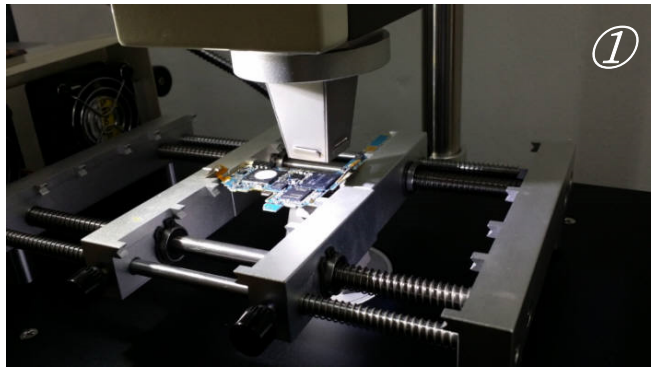


H/W-based acquisition methods(2)

- Chip-off-based acquisition method
  - Physically removes flash memory chips on PCB
    - Can acquire all flash memory data
    - Use in limited situations because it separates the flash memory

Chip-off-based acquisition method



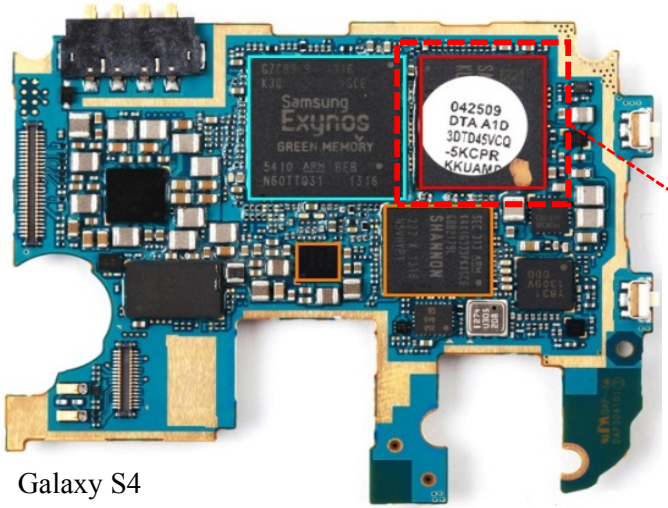


BACKGROUND

Flash memory of smart phones

- Flash memory is used mainly to store data
  - Widely use in embedded devices such as mobile devices
  - Recently use the embedded Multi-Media Card(eMMC)
    - Integrate NAND flash and a controller into a package
    - Usually use the EXTended file system4(EXT4)
    - Mount and operate partitions such as BOOT, RECOVERY, SYSTEM, and USERDATA

Flash memory of smart phones



Galaxy S4

Partition information table
Bootloader partition
Boot partition
Recovery partition
Modem partition
FOTA partition
Cache partition
System partition
User data partition

Firmware update of Android smart phones

- All Android smart phones use firmware update
  - Update the Android OS or patch S/W problems
  - Each manufacturer provides their own firmware update programs
    - Google FASTBOOT, Samsung Kies, Odin, LG Software & tools download, Pantech Self-Upgrade, HTC Sync Manager, Sony PC Companion, Xiaomi MiFlash, etc
- Android firmware update
  - Firmware update only run in the special mode
    - The bootloader and USB function can only operate (Emergency mode)
  - Only way to access the flash memory directly by S/W
    - Not only the write commands for flashing but also **the read commands for flash memory**

Need to analyze firmware update protocols to acquire data of flash memory

ANDROID PHYSICAL ACQUISITION BASED ON FIRMWARE UPDATE PROTOCOLS

1. Firmware update protocols

2. Analysis of LG firmware update protocols

3. Analysis of Pantech firmware update protocols

4. Analysis of Samsung firmware update protocols

Android firmware update process

Firmware update process

- Partition information table
- Bootloader partition
- Boot partition
- Recovery partition
- Modem partition
- FOTA partition
- Cache partition
- System partition
- User data partition

GPT Partition information.

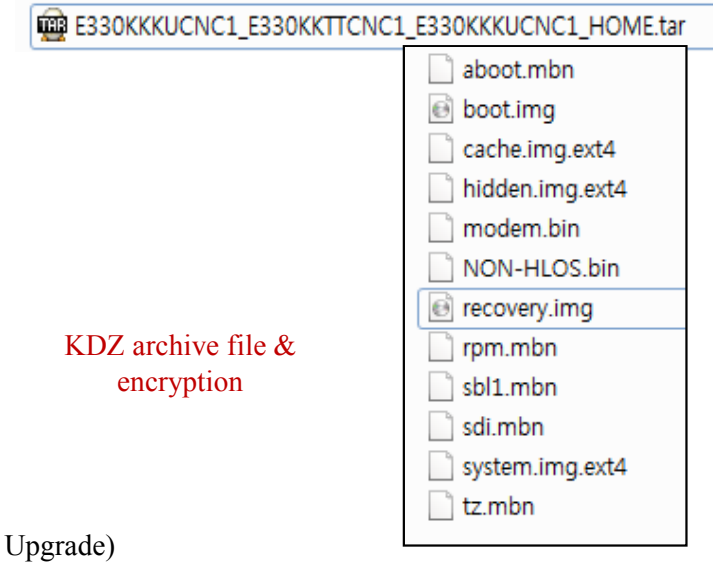
Idx	Part Name	Start TOT	Start DEV	End
0	modem	32768	-	163839
1	sbl1	163840	-	165887
2	dbi	165888	-	166911
3	DDR	196608	-	197631
8	pad	296960	-	296961
9	modemst1	327680	-	333823
10	modemst2	333824	-	339967
11	pad1	339968	-	339969
12	misc	360448	-	393215
13	persist	393216	-	458751
14	recovery	458752	-	491519
15	fsg	491520	-	497663
16	fsc	524288	-	525311
17	ssd	525312	-	526335
18	pad2	526336	-	526337
19	encrypt	526338	-	527361
20	drm	557056	-	573439
21	sns	573440	-	589823
22	ve	589824	-	622591
23	laf	622592	-	688127
24	fota	688128	-	753663
25	mpt	753664	-	819199
26	carrier	819200	-	884735
27	eri	884736	-	901119
28	blus	901120	-	905215
29	dbibak	905216	-	906239
30	rpmbak	906240	-	908287
31	tzbak	908288	-	910335
32	rct	910336	-	910351
33	system	917504	-	7733247
34	cache	7733248	-	9371647
35	tombstones	9371648	-	9535487
36	spare	9535488	-	9568255
37	userdata	9568256	-	61046783
38	grow	61046784	-	61071326

1. Analyze the GPT partition

2. Write the partition

SAMSUNG (ODIN)

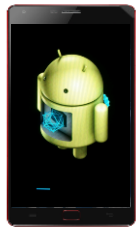
TAR archive file



LG (DL)

KDZ archive file & encryption

PANTECH (SW Upgrade)



AES-CBC & ZIP archive file

Is there the read command for dumping the flash memory ?

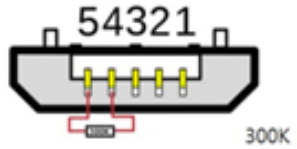
Firmware update mode

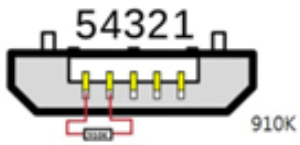
- To update firmware or acquire data by accessing flash memory
  - Should be in the firmware update mode rather than the normal boot mode
    - Only the bootloader and USB module are activated

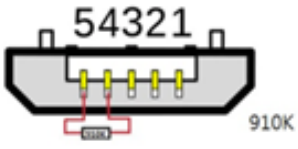
The integrity can be preserved to dump an image of the flash memory

Methods used for entering the firmware update mode

Model	Mode name	Key combination
Samsung Galaxy	ODIN	Press and hold, at the same time, Volume Down, Home and Power key (then press Volume Up key) or Connect 300K ohm USB Jig
LG Optimus	DOWNLOAD	Press and hold Volume Up key + plug in the phone into a computer using a microUSB cable or Connect 910K ohm USB Jig
Pantech Vega	PDL download	Press and hold, at the same time, Volume Up, Volume Down, Home, and Power key
Google Nexus 4/5	DOWNLOAD	Press and hold Volume Up key + plug in the phone into a computer using a microUSB cable or Connect 910K ohm USB Jig



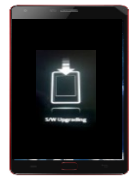




SAMSUNG

LG

PANTECH



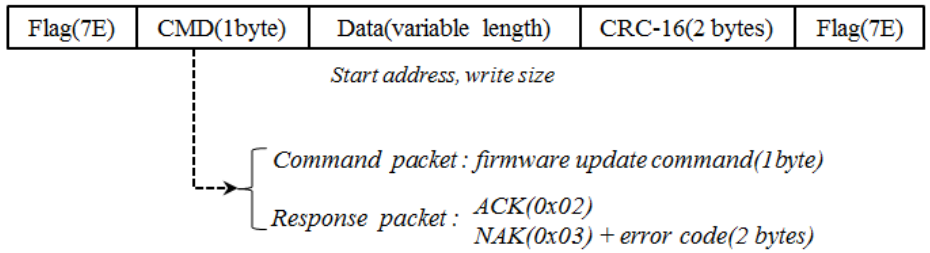
LG firmware update protocols(1)

LG firmware update commands

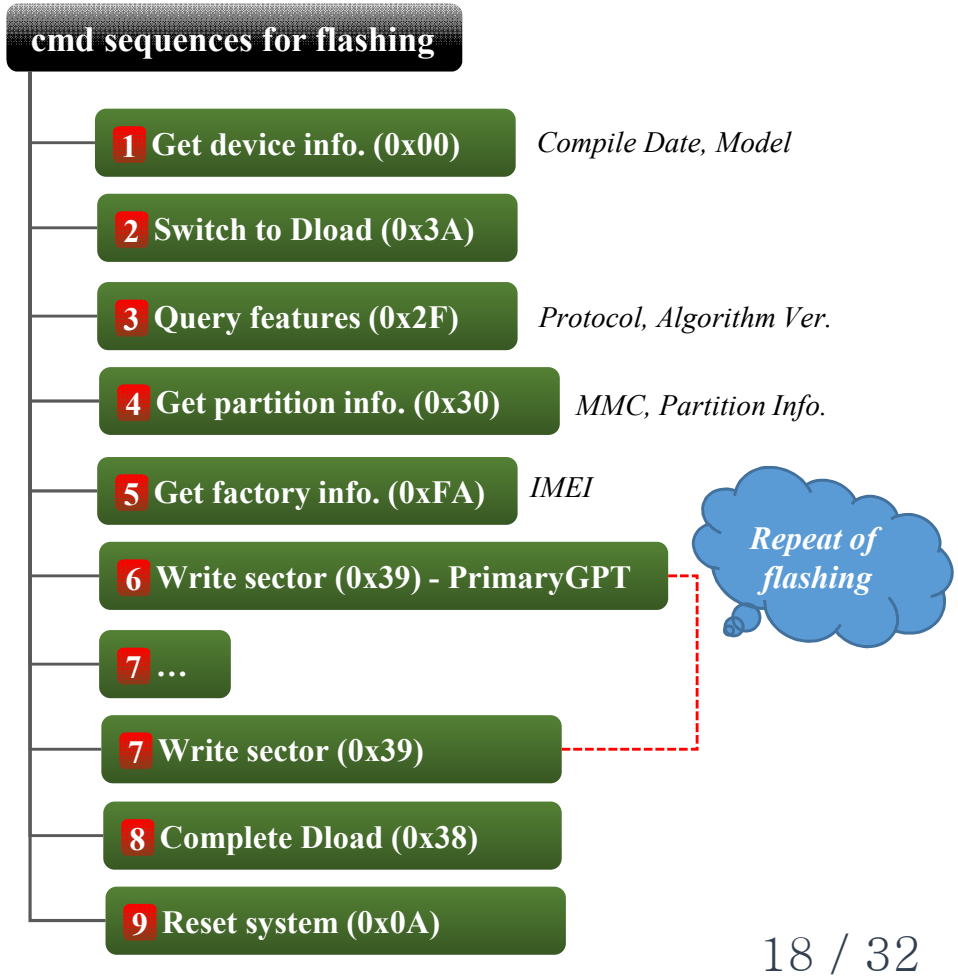
- Implement in the bootloader(SBL3 or ABOOT)
- High-Level Data Link Control(HDLC) frame structure

LG firmware update command

Packet format (HDLC frame)



Command	Description
0x00	Get device information (model, compile date)
0x3A	Go download mode
0x2F	Get protocol version, algorithm version
0x30	Get MMC and Partition information
0xFA	Get factory information (IMEI, Mac address)
0x12	Read RAM memory
0x39	Write flash memory
0x0A	Reset system

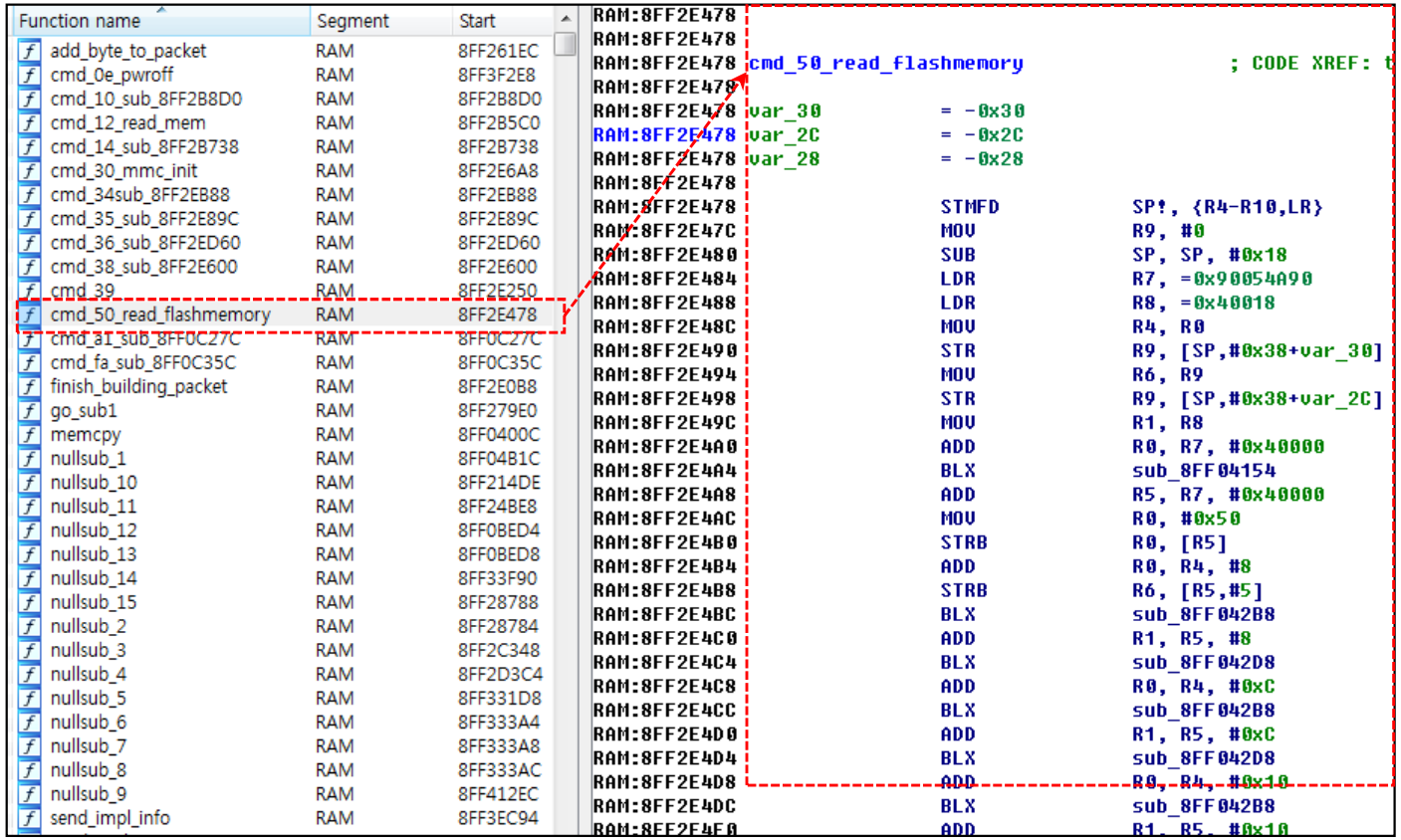


LG firmware update protocols(2)

LG flash memory dump command

- We find the read command for flash memory by reverse engineering the bootloader
  - Analyze the firmware update command by using IDA Pro

Reverse engineering of the LG SBL3 bootloader



LG firmware update protocols(3)

LG flash memory dump command

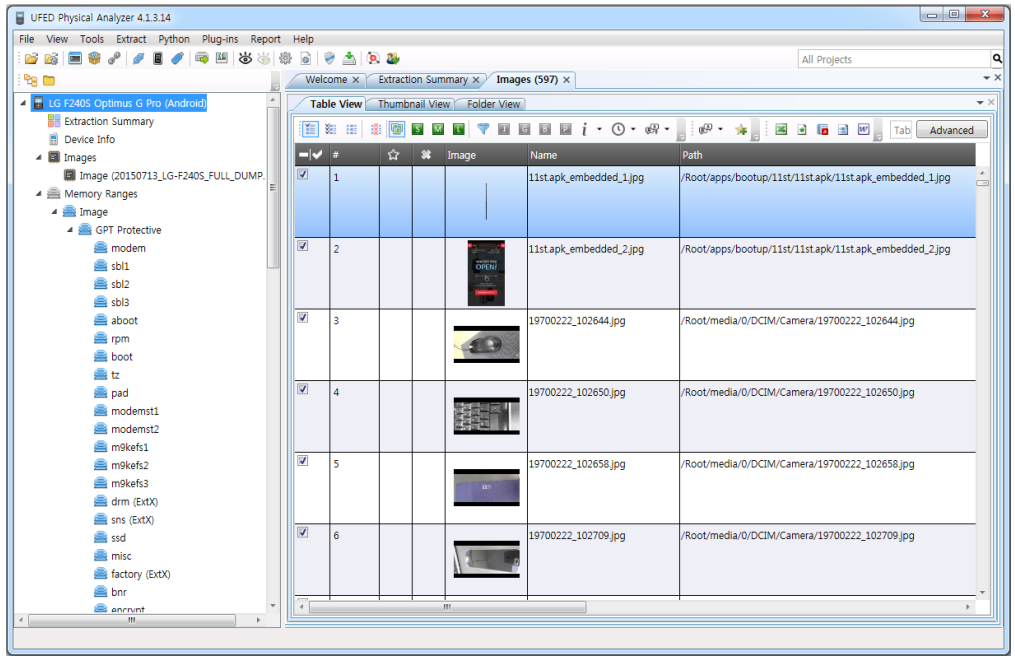
Read command format

0x50	Sub command	Start address(4)	Dump size(4)	CRC

Acquisition capture (LG-F240S)

```
Send 0x1c bytes to the device
 7E 50 01 01 00 00 00 00 00 00 80 02 00 00 02 00   ~P........€.....
 00 00 00 00 00 00 00 00 00 00 7C 1B 7E            ..........|.~      PC -> Phone
000003: Bulk or Interrupt Transfer (UP), 26.01.2015 15:07:07.077 +0.
Pipe Handle: 0x123046b8 (Endpoint Address: 0x83)
Get 0x3f454 bytes from the device
 50 01 01 00 00 00 00 00 00 80 02 00 00 02 00 00   P........€......
 04 E7 03 00 00 00 00 00 E0 1C 41 4E 44 52 4F 49   .?.....?ANDROID!
 44 21 A0 00 74 00 00 80 20 80 63 9C 29 00 00 00   ?.t..€ €c?... ?.
 20 82 00 00 00 00 00 00 10 81 00 01 20 80 00 08   ......?.. €.....
 00 00 00 00 00 00 38 06 FA E0 89 76 6D 61 6C 6C
 6F 63 3D 36 30 30 4D 20 63 6F 6E 73 6F 6C 65 3D   0M console=ttyHS
 74 74 79 48 53 4C 30 2C 31 31 35 32 30 30 2C 6E   L0,115200,n8 lpj
 38 20 6C 70 6A 3D 36 37 36 37 37 20 75 73 65 72   =67677 user_debu
 5F 64 65 62 75 67 3D 33 31 20 6D 73 6D 5F 72 74   g=31 msm_rtb.fil
 62 2E 66 69 6C 74 65 72 3D 30 78 30 20 65 68 63   ter=0x0 ehci-hcd
 69 2D 68 63 64 2E 70 61 72 6B 3D 33 20 63 6F 72   .park=3 coresigh
 65 73 69 67 68 74 2D 65 74 6D 2E 62 6F 6F 74 5F   t-etm.boot_enabl
 65 6E 61 62 6C 65 3D 30 20 61 6E 64 72 6F 69 64   e=0 androidboot.
 62 6F 6F 74 2E 68 61 72 64 77 61 72 65 3D 67 65   hardware=geehrc8   Phone -> PC
 65 68 72 63 38 AF FC 38 13 F0 FF F0 28 E0 04 5A
 36 5C F0 DC 20 32 3A 7D 5D 07 D5 74 26 47 FE D5
 DD D3 D4 BC 39 68 F0 FF F0 2B 39 51 F0 FF F0 FF
 F0 FF F0 FF F0 04 98 A0 E1 04 F0 06 EA 02 00 00
 EA 18 28 6F 01 00 00 00 26 E0 1B 00 74 00 01 70
 A0 E1 02 80 A0 E1 00 20 0F E1 03 00 12 E3 01 00
 00 1A 17 00 A0 E3 56 34 12 EF 00 20 0F E1 C0 20
 82 E3 02 F0 21 E1 2C B2 E0 EC A4 47 9F E5 55 00
 00 EB 4A 0F 8F E2 4E 1C 90 E8 1C D0 90 E5 01 00
```

Verify the dumped image (LG-F240S)

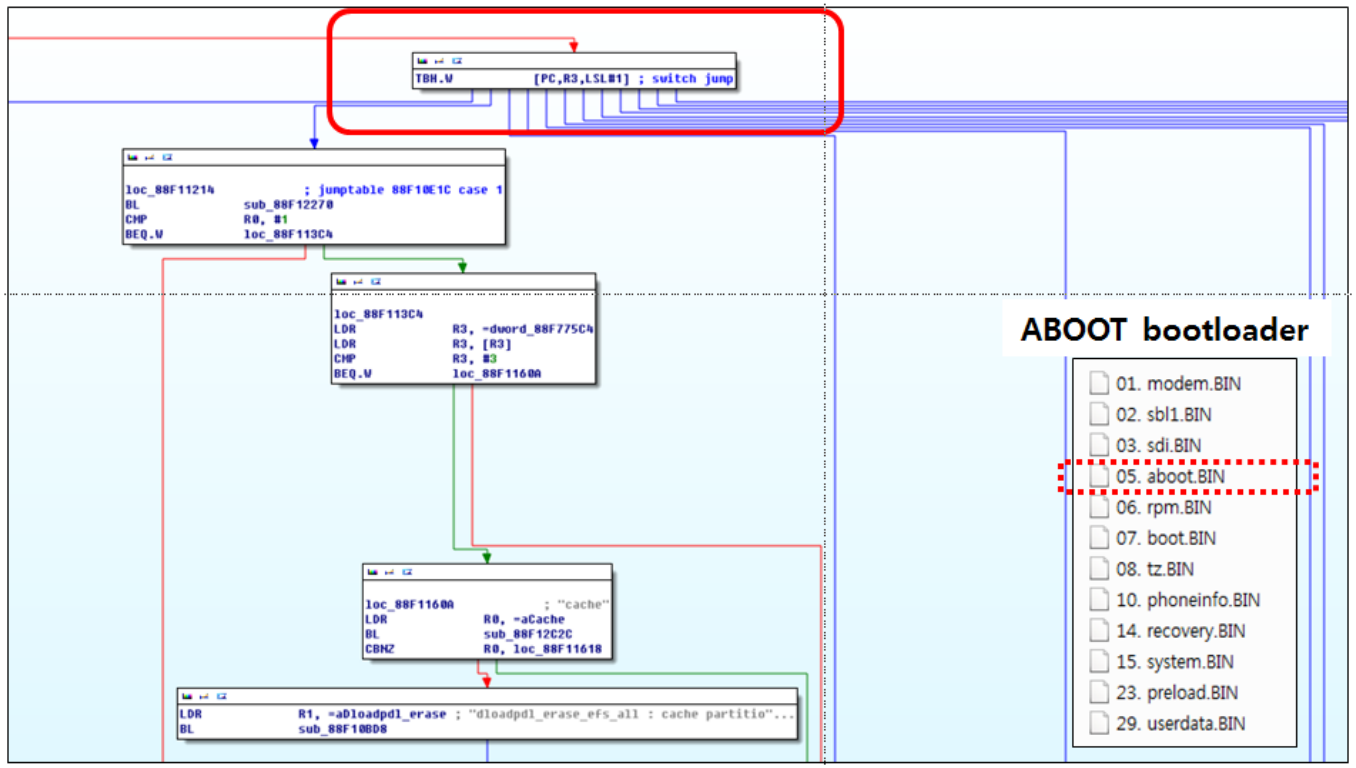


Can be analyzed through Cellebrite UFED Physical Analyzer

Pantech firmware update protocols(1)

- Pantech firmware update commands
  - Implement in the bootloader(SBL3 or ABOOT)
  - Analyze the firmware update commands by using IDA Pro

Reverse engineering of ABOOT bootloader



Pantech firmware update commands

Command	Description
0x00	Firmware update ready command
0x01	Reset command
0x02	Write ready command
0x03	Write complete command
0x04	Partition erase command
0x05	Partition write command

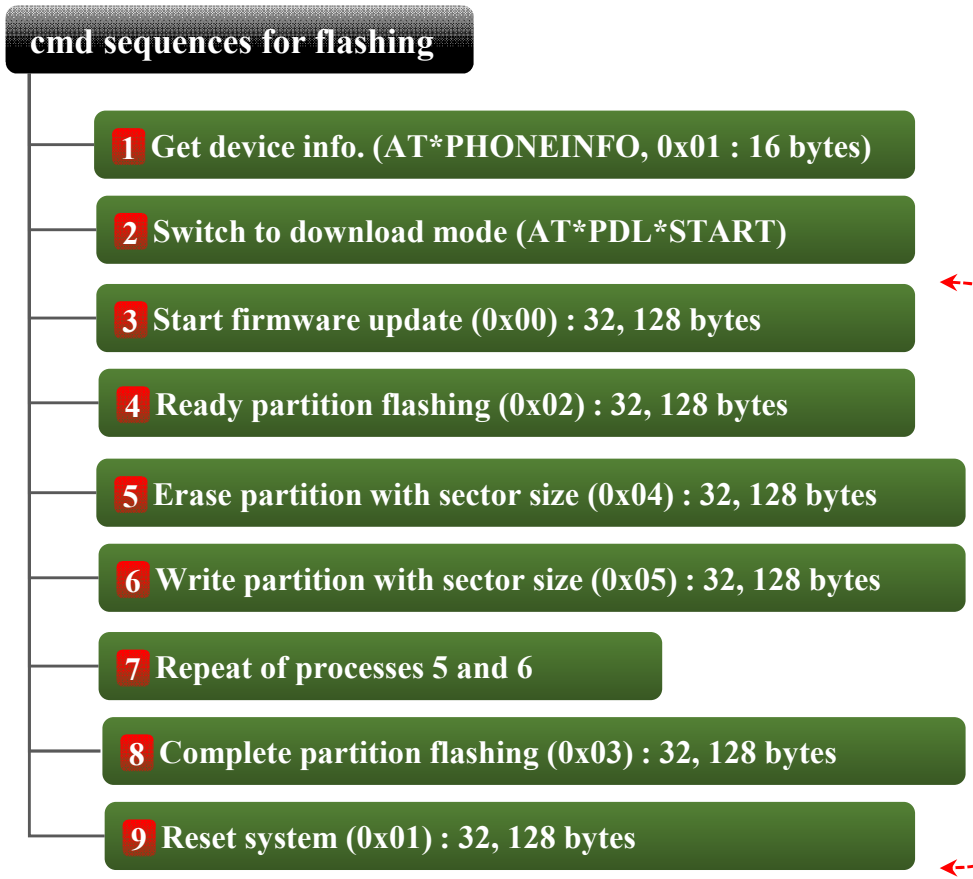
Pantech firmware update protocols(2)

Pantech firmware update commands

The length of the command packet

- 32-byte command packet was used up to the Vega Iron
- 128-byte command packet was used in subsequent models

Pantech firmware update process



32-byte command packet

cmd(4)	data(28)			
	Partition ID (4)	Start address (4)	End address (4)	.....

128-byte command packet

cmd(4)	data(124)				
	Partition ID (4)	Partition name (12)	Start address (8)	End address (8)	.....

Pantech firmware update protocols(3)

Pantech flash memory dump command

- Can acquire using the read command for flash memory (0x06)

Read command format

Read command (32 bytes)

0x06(4)	Partition ID(4)	0x00(4)	Start address(4)	Dump size(4)	0x00(12)

command	partition ID		start address
06 00 00 00	0A 00 00 00	00 00 00 00	00 00 00 00
00 02 00 00	00 00 00 00 00 00 00 00 00 00 00 00		

dump size

Read command (128 bytes)

0x06(4)	Partition ID(4)	Partition name(12)	0x00(8)	Start address(8)	Dump size(8)	0x00(84)

command	partition ID	partition name(=phoneinfo)	
06 00 00 00	0A 00 00 00	70 68 6F 6E 65 69 6E 66	
6F 00 00 00	00 00 00 00	00 00 00 00	start address
00 00 00 00	00 00 08 00	00 00 00 00	dump size
00 00 00 00	00 00 00 00	00 00 00 00	
00 00 00 00	00 00 00 00	00 00 00 00	
00 00 00 00	00 00 00 00	00 00 00 00	
00 00 00 00	00 00 00 00	00 00 00 00	
00 00 00 00	00 00 00 00	00 00 00 00	

Acquisition capture (IM-A910S)

Acquisition capture

```
Send 0x80 bytes to the device
06 00 00 00 07 00 00 00 42 4F 4F 54 00 00 00 00   ........BOOT....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 10 01 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
                                                          PC -> Phone
011124: Bulk or Interrupt Transfer (UP), 26.01.2015 16:18:01.515 +0.0. (1. 
Pipe Handle: 0x134324b8 (Endpoint Address: 0x81)
Get 0x8000 bytes from the device
06 00 00 00 07 00 00 00 42 4F 4F 54 00 00 00 00   ........BOOT....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 10 01 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
41 4E 44 52 4F 49 44 21 60 6E 61 00 00 80 00 00   ANDROID!`na..€..
DA CC 0E 00 00 00 00 01 00 00 00 00 00 00 F0 00                    Phone -> PC
00 01 00 00 00 08 00 00 00 68 03 00 00 00 00 00   ........h.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
63 6F 6E 73 6F 6C 65 3D 4E 55 4C 4C 2C 31 31 35   onsole=NULL,1152
32 30 30 2C 6E 38 20 61 6E 64 72 6F 69 64 62 6F   00,n8 androidboo
6F 74 2E 68 61 72 64 77 61 72 65 3D 71 63 6F 6D   t.hardware=qcom
20 75 73 65 72 5F 64 65 62 75 67 3D 33 31 20 6D    user_debug=31 m
73 6D 5F 72 74 62 2E 66 69 6C 74 65 72 3D 30 78   s m_rtb.filter=0x
33 37 20 65 68 63 69 2D 68 63 64 2E 70 61 72 6B   3 7 ehci-hcd.park
3D 33 20 76 6D 61 6C 6C 6F 63 3D 32 36 30 4D 20   = 3 vmalloc=260M 
6C 6F 67 6C 65 76 65 6C 3D 30 00 00 00 00 00 00   loglevel=0.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Samsung firmware update protocols(1)

Samsung firmware update commands

- Implement in the bootloader(SBOOT.bin)

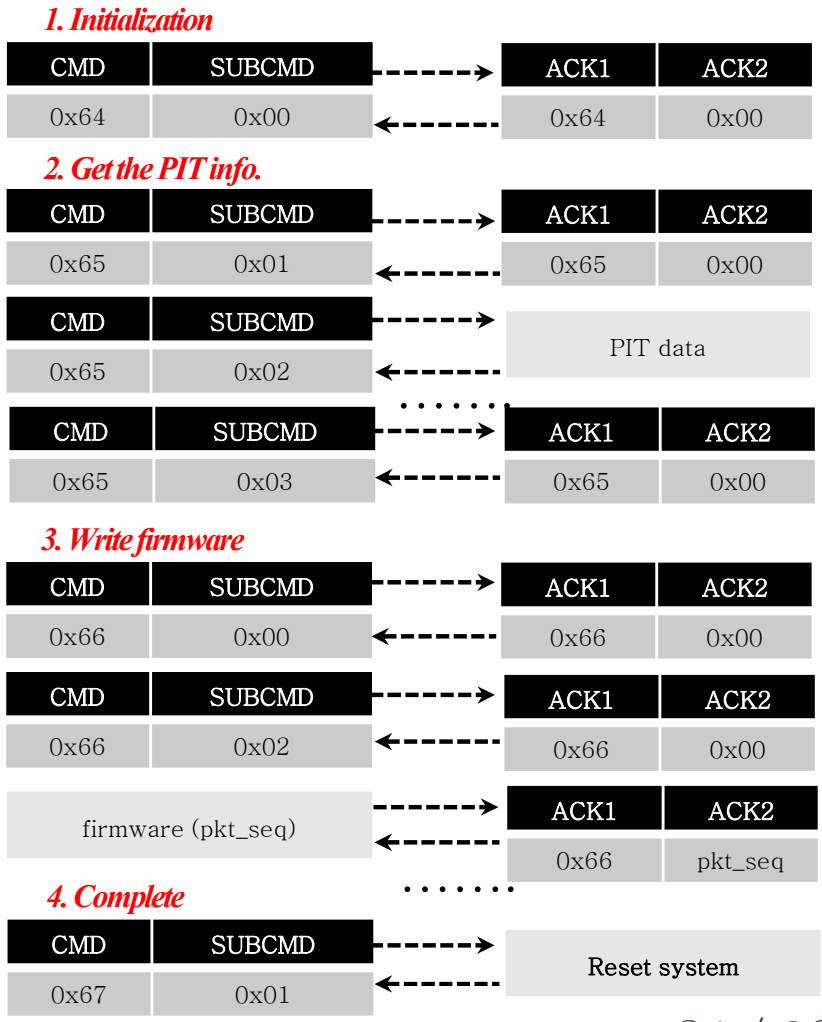
Samsung firmware update command

CMD (4Bytes)	DATA(1020Bytes)		
	SUBCMD(4Bytes)	PARAM1(4Bytes)	…

ACK1 (4Bytes)	ACK2 (4Bytes)

Command	Sub commands	Description	
0x64	0x00	Firmware update protocol initialization command	-
0x65	0x00	Partition information table write ready command	-
	0x01	Partition information table read ready command	-
	0x02	Partition information table write/read command	-
0x66	0x00	Write ready command for flash memory	-
	0x01	Read ready command for flash memory	Code removal
	0x02	Write command for flash memory	-
	0x03	Read command for flash memory	Code removal
0x67	0x01	Firmware update complete command	-

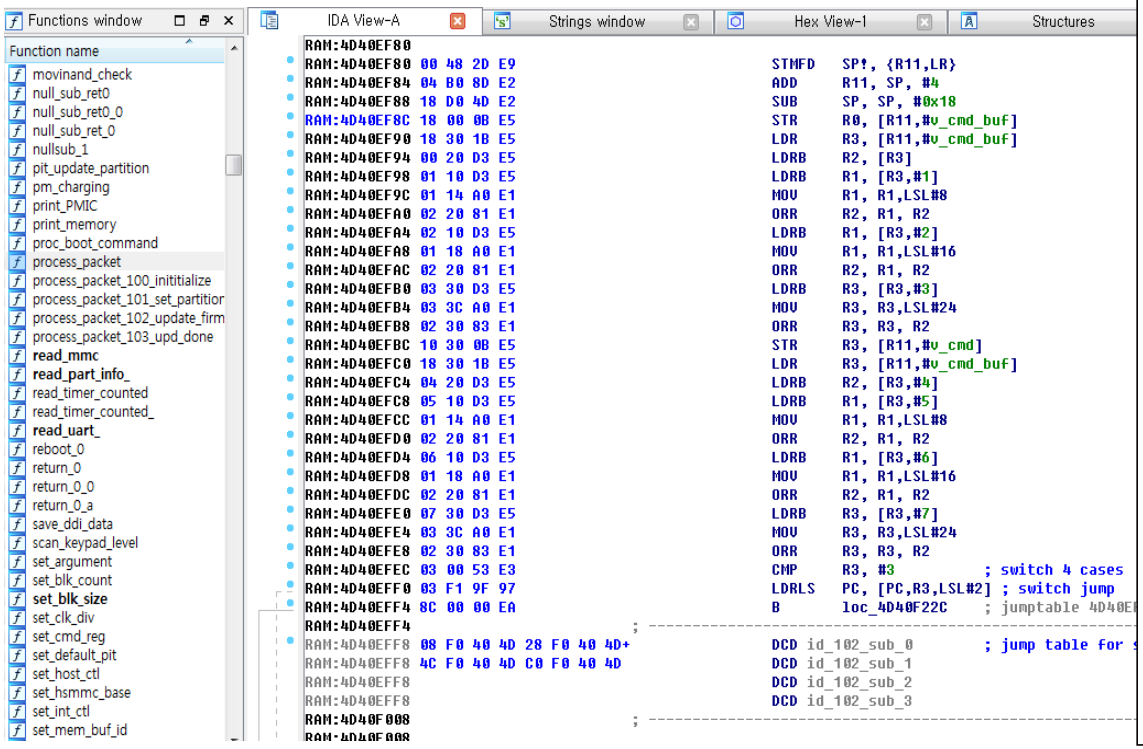
Firmware update process



Samsung firmware update protocols(2)

- Samsung flash memory dump command
  - Read command(CMD:0x66, SUBCMD: 0x01, 0x03)
    - The actual read code had been removed
      - The smartphone only sends an ACK message without data
    - **Additional research for patching the read command is required**

Analyze the bootloader by using IDA Pro




```
int __fastcall process_packet_102_update_firmware(int result)
{
  int v_cmd_buf; // [sp+4h] [bp-18h]@1
  int v2; // [sp+8h] [bp-14h]@3
  int v_cmd; // [sp+Ch] [bp-10h]@1
  int v_binary_phone; // [sp+10h] [bp-Ch]@10
  int v_status; // [sp+14h] [bp-8h]@13

  v_cmd_buf = result;
  v_cmd = (*(_BYTE *)(result + 3) << 24) | (*(_BYTE *)(result + 2) << 16) | (*(_BYTE *)
  switch ( (*(_BYTE *)(result + 7) << 24) | (*(_BYTE *)(result + 6) << 16) | (*(_BYTE *
  {
    case 0:
      id_102_flag = 0;
      result = upload_ack(v_cmd, 0);
      break;
    case 1:
      id_102_flag = 1;
      result = upload_ack(v_cmd, v2);
      break;
    case 2:
      if ( id_102_flag != 1 && !id_102_flag )
      {
        v2 = (*(_BYTE *)(result + 11) << 24) | (*(_BYTE *)(result + 10) << 16) | (*(_BY
        upload_ack(v_cmd, 0);
        result = download_data(v2);
      }
      break;
    case 3:
      if ( id_102_flag == 1 )
      {
        result = upload_ack(v_cmd, 0);
      }
      else if ( !id_102_flag )
      {
        v_binary_phone = (*(_BYTE *)(result + 11) << 24) | (*(_BYTE *)(result + 10) <<
        if ( !dit.set_nps_update )
```

Code removal of the read command (CMD: 0x66, SUBCMD: 0x01)

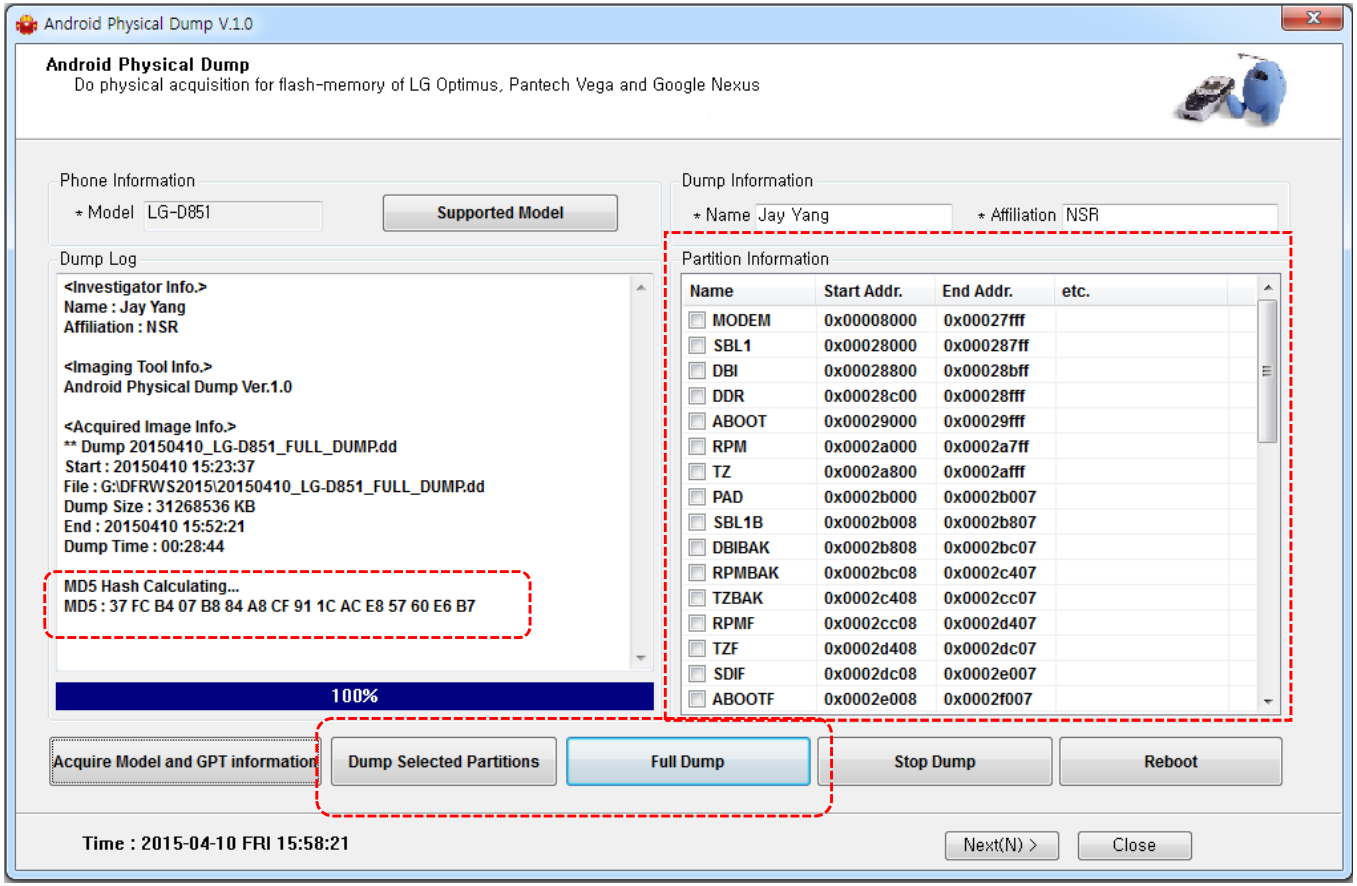
Code removal of the read command (CMD: 0x66, SUBCMD: 0x03)

ANDROID PHYSICAL DUMP (APD)

The APD tool

- Physical dump tool for Android smart phones
  - Currently support over 80 of the latest Android models
    - LG Optimus, Pantech Vega, and Google Nexus
    - Physical dump after booting with firmware update mode
    - Support a partition dump and a whole flash memory dump

Android Physical Dump (APD)



EXPERIMENTS WITH APD

Experiments with APD(1)

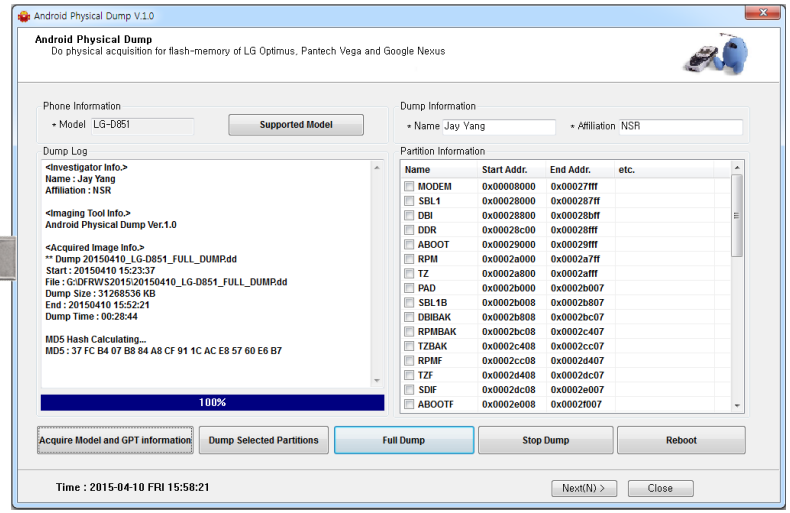
Experimental method

Experimental factors

- Preserving the integrity of the dumped image
- Acquisition speed
- Physical acquisition from screen-locked smart phones

Compare the APD tool with existing acquisition methods

- Cellebrite UFED 4PC, Dump method based on a custom recovery image, ADB physical via rooting exploitations, JTAG based acquisition



Experiments with APD(2)

- Experimental results
  - Preserving the integrity of the dumped image
    - Compare the hash values of the dumped images after acquiring five times
  - Acquisition speed
    - Take about 30 min on average to acquire the 32 GB memory
  - Physical acquisition from screen-locked smart phones
    - Execute acquisition after turning off the phone and rebooting in the firmware update mode

Experimental results

Item	Proposed method (APD)	Cellebrite UFED 4PC	Custom recovery mode dump	ADB dump using root exploitation	JTAG-based acquisition
Integrity guaranteed	O	X	X	X	O
Acquisition speed (size: 32 GB)	30 min	120 min	120 min	180 min	480 min [a]
Dump of smartphone with screen-lock	O	X	X	X	O

[a] *JTAG-based acquisition method excludes the disintegration and connection time*

CONCLUSION

Conclusion

- Propose new acquisition method for Android smart phones
  - Analyze the firmware update protocols of smart phones
    - LG and Pantech models
      - Acquire a whole flash memory data by using the read command
    - Samsung models
      - Additional research for patching the read command is required
  - Compare the APD tool with existing acquisition methods
    - Guarantee the integrity of the entire flash memory
    - Acquire at a high speed
    - Available dump from screen-locked smart phones
  - Necessary to analyze the firmware update protocol whenever new Android smart phones are launched
    - Physical acquisition can be performed for all models of the manufacturers as long as the acquisition method by analyzing the firmware update protocol is found

Q & A