Mobile Forensic

Rahmaa #exploit0x3

Evidence / Artifact

~ EMMC/EMCP (Flash Memory)
  -- Internal Memory
~ Volatile Memory (RAM)
~ SD Card
~ SIM Card/UICC

Acquisition Type

~ Phisical Acquisition
~ Logical Acquisiton

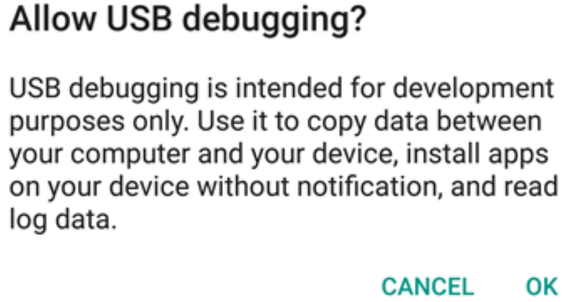
Physical Acquisition

Physical acquisition is a bit-by-bit copy of the physical storage and it can be invasive (JTAG, ISP or Chip-off) or non-invasive, with the use of “dd” command

Chipp Off

JTAG

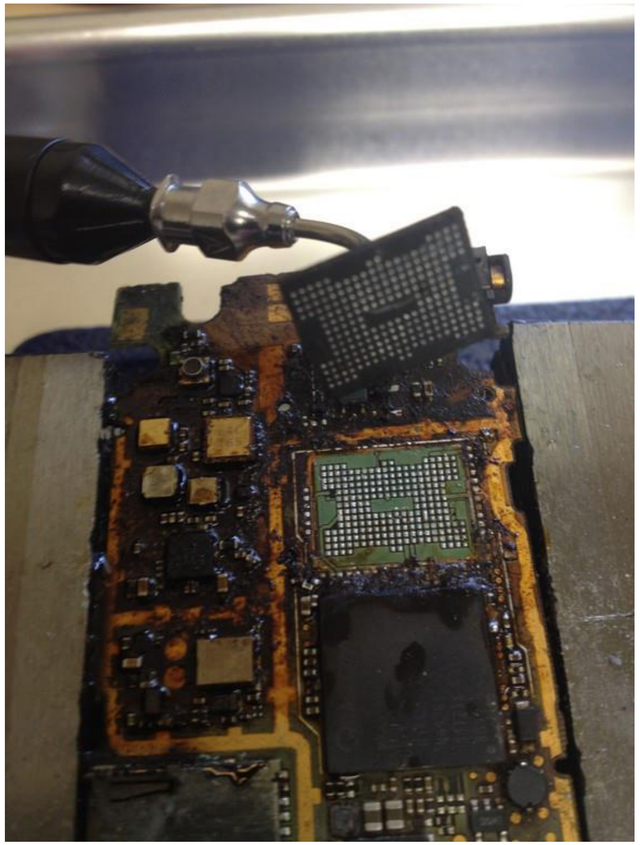
ISP



Full partition #root access with dd

Chip-Off

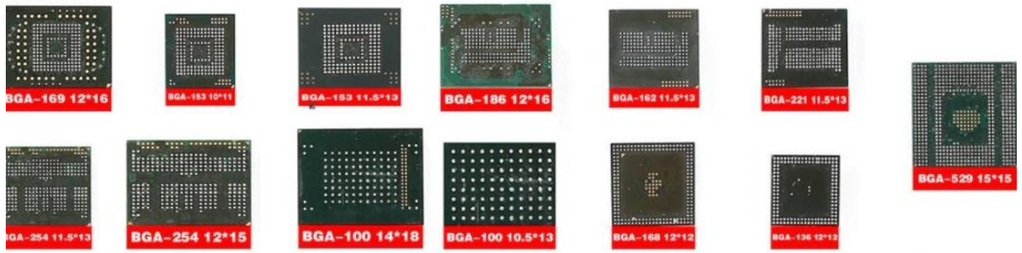
- Acquire damaged and broken device
- Removal of the memory chip (emmc/emcp)



Acquisition Flow

EMMC Socket adapter/reader → Flash Box → Dump data

--> RAW Image : .dd, .raw, etc



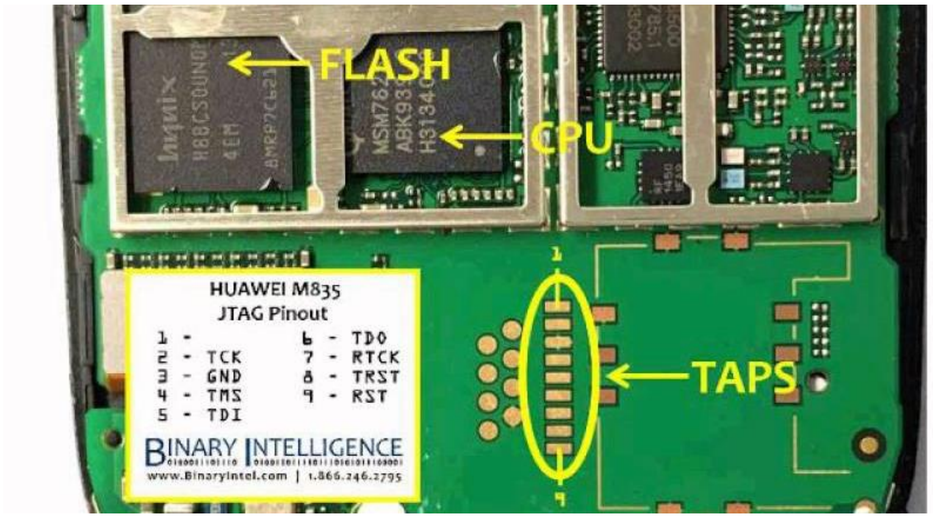
1. EMMC

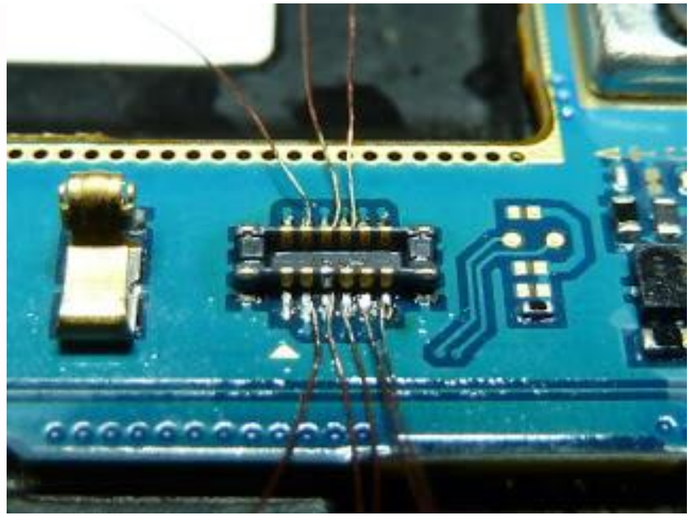
2. emmc BGA socket

3. Flash Box : RIFF, UFI Box, Z3X Easy JTAG

JTAG (Join Test Action Group)

- Requires a high skill level , disassembling the device (can be invasive invasive) --> Identification **TAP (Test Access Ports)**
- Slow acquisition speed
- Device must be powered on



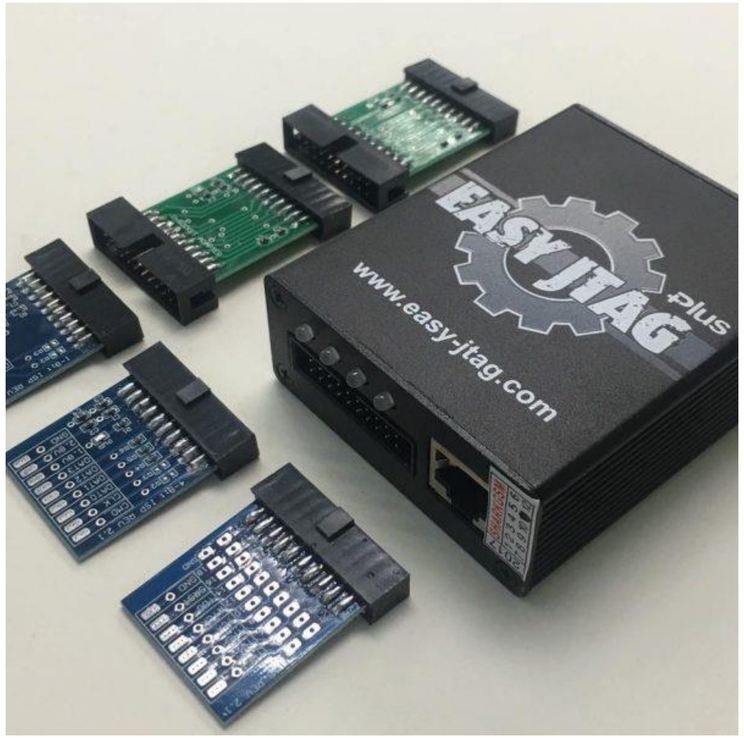


Acquisition Flow

JTAG TAP → PCB (via Solder, molex, jig) → JTAG Box → Dump data

--> RAW Image : .dd, .raw, etc

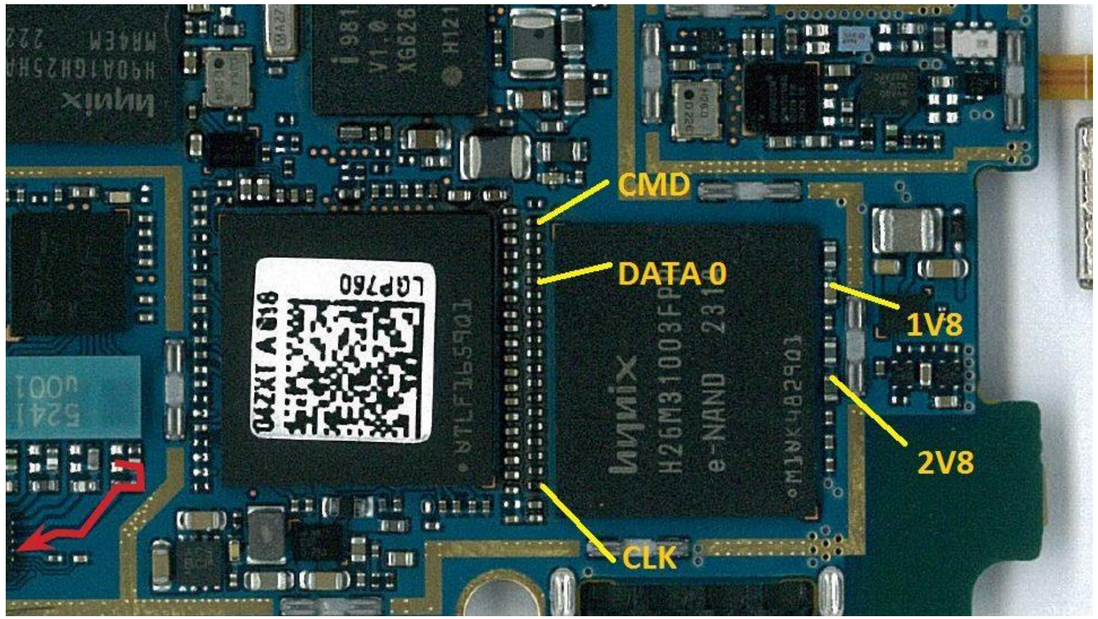
1. Connecting JTAG TAP to PCB / JTAG Adapter

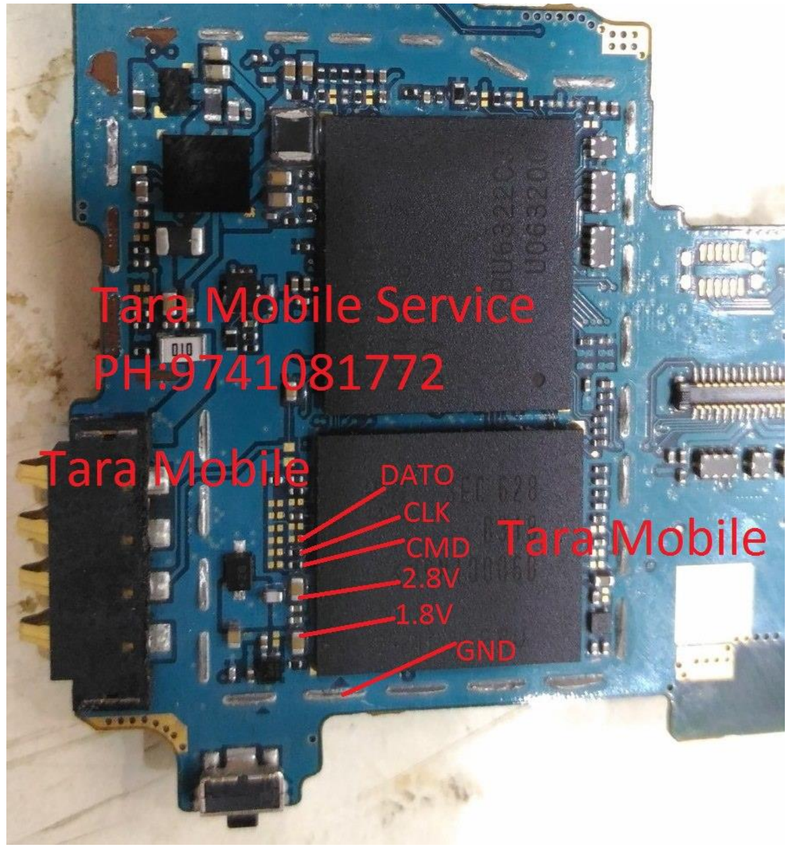


2. Dump using JTAG Box : RIFF, UFI Box, Z3X Easy JTAG

ISP (In-system programming)

- Identification **ISP Pinout / EMMC TAP.** ISP connect directly to the eMMC/eMCP flash memory, it bypasses the processor == is faster than JTAG
- Device doesn't need to be on

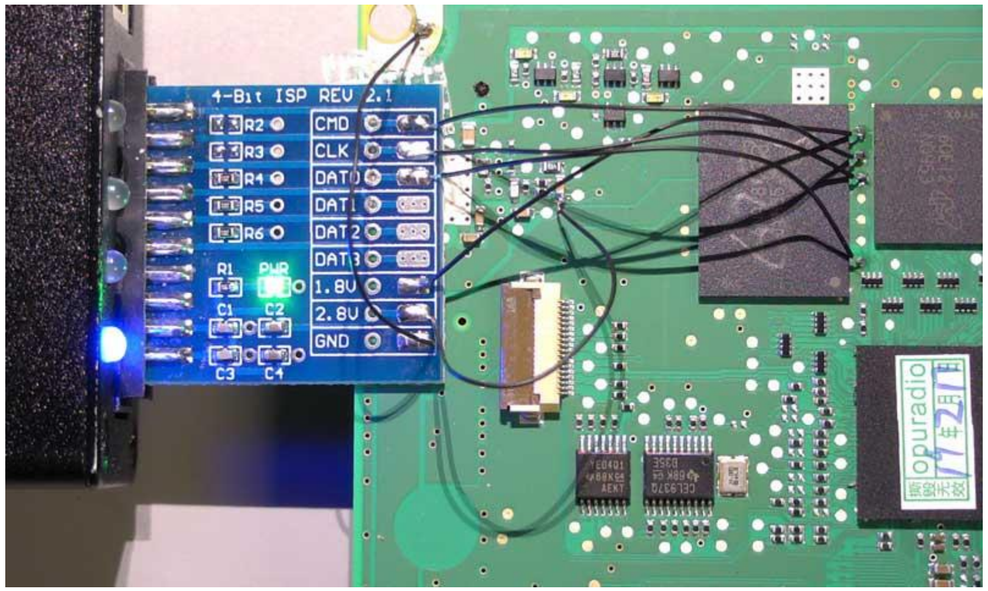




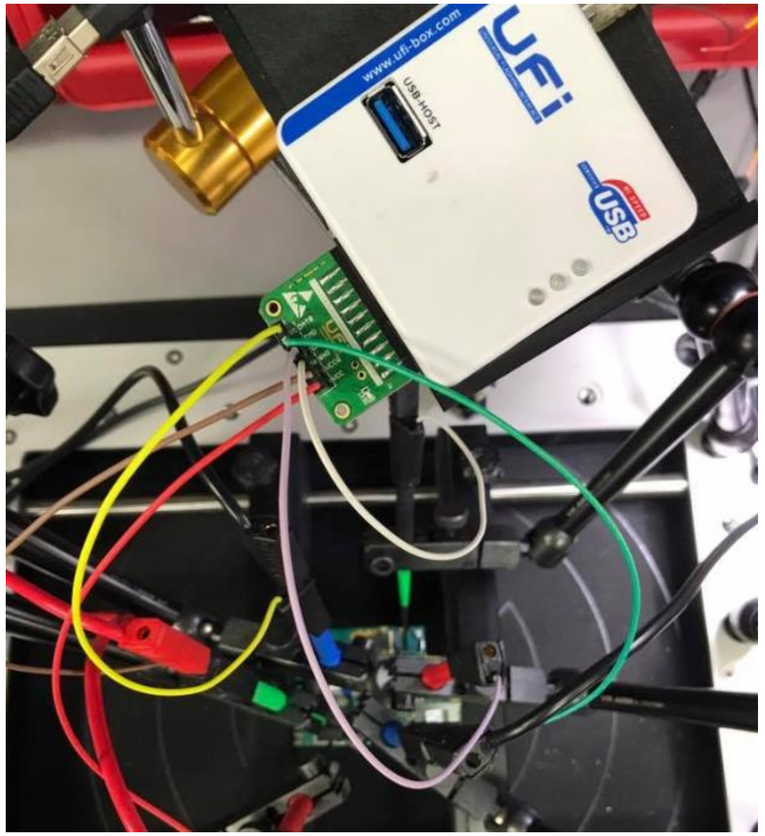
Acquisition Flow

Direct EMMC Pinout → Socket adapter → Flash Box → Dump data

--> RAW Image : .dd, .raw, etc



1. connect eMMC TAPs with a flash box/socket adapter



2. Flash Box : AFT Box, UFI Box, eMMC Pro

#root access

➤ Via **USB Debugging , Tool : dd**

> cat **/proc/dumchar_info** : Informasi partisi

- MTD Partition Layout > **/proc/mtd**
- EMMC Partition Layout > **/proc/emmc**
- MMC Partition Layout > **/proc/partitions**

Acquisition Flow

USB Debugging : On --> Full Partition atau /dev/block/<userdata>

C:\WINDOWS\system32\cmd.exe - adb shell

```
C:\Users\rahmaa>adb shell
shell@android:/ $ su
root@android:/ # cat /proc/dumchar_info
Part_Name       Size    StartAddr       Type    MapTo
preloader    0x0000000000600000   0x0000000000000000   2   /dev/misc-sd
mbr          0x0000000000080000   0x0000000000000000   2   /dev/block/mmcblk0
ebr1         0x0000000000080000   0x0000000000080000   2   /dev/block/mmcblk0p1
pro_info     0x0000000000300000   0x0000000000100000   2   /dev/block/mmcblk0
nvram        0x0000000000500000   0x0000000000400000   2   /dev/block/mmcblk0
protect_f    0x000000000a00000    0x0000000000900000   2   /dev/block/mmcblk0p2
protect_s    0x000000000a00000    0x0000000001300000   2   /dev/block/mmcblk0p3
seccfg       0x0000000000020000   0x0000000001d00000   2   /dev/block/mmcblk0
uboot        0x0000000000060000   0x0000000001d20000   2   /dev/block/mmcblk0
bootimg      0x0000000000600000   0x0000000001d80000   2   /dev/block/mmcblk0
recovery     0x0000000000600000   0x0000000002380000   2   /dev/block/mmcblk0
sec_ro       0x0000000000040000   0x0000000002980000   2   /dev/block/mmcblk0
misc         0x0000000000080000   0x00000000029c0000   2   /dev/block/mmcblk0
logo         0x0000000000300000   0x0000000002a40000   2   /dev/block/mmcblk0
expdb        0x000000000a00000    0x0000000002d40000   2   /dev/block/mmcblk0
android      0x000000001f400000   0x0000000003740000   2   /dev/block/mmcblk0p4
cache        0x0000000006400000   0x0000000022b40000   2   /dev/block/mmcblk0p5
usrdata      0x000000002bc00000   0x0000000028f40000   2   /dev/block/mmcblk0p6
fat          0x0000000019de0000   0x0000000054b40000   2   /dev/block/mmcblk0p7
bmtpool      0x0000000001500000   0x00000000ff9f00a8   2   /dev/block/mmcblk0
Part_Name:Partition name you should open;
Size:size of partition
StartAddr:Start Address of partition;
Type:Type of partition(MTD=1,EMMC=2)
```

1. Identification Partition

shell@android:/# **cat /proc/dumchar_info**
shell@android:/# **cat /proc/emmc**

C:\WINDOWS\system32\cmd.exe - adb shell

```
root@android:/ #
root@android:/ #
root@android:/ # dd if=/dev/block/mmcblk0p6 | busybox nc -l -p 8888
```

C:\WINDOWS\system32\cmd.exe - ncat 127.0.0.1 8888

```
E:\dfir>
E:\dfir>ncat 127.0.0.1 8888 > android_data.dd
```

2. Imaging with dd

root@rahma:/# **adb forward tcp:8888 tcp:8888**
shell@android:/# **dd if=/dev/block/<userdata> | busybox nc -l -p 8888**
root@rahma:/# **nc 127.0.0.1 8888 > userdata.dd**

shell@android:/# **dd if=/dev/block/block of=sdcard/image.img --> adb pull sdcard/image.img**

Physical Acquisition?

- Acquire damaged and broken device (Chip Off, ISP)
- Bit-by-bit copy of all of its memory (unallocated space)
- Recover deleted data
- Analyze slack space
- Bypass lock screen (fingerprint, face unlock passcode, password vs)
- Full keychain extraction (Decrypt WhatsApp chat backup : msgstore.db.crypt) -- */data/data/com.whatsapp/files/key*

Cellebrite UFED touch

http://www.onretrieval.com/images/PDF_s/Cellebrite_OnRetrieval_UFED_Touch_manual_usuario.pdf

Glossary

https://www.cellebrite.com/en/blog/overcoming-locked-android-powered-devices/

Logical Acquisition

Logical acquisition is a bit-by-bit image acquisition of logical storage objects that reside on

~ ADB Backup
~ ADB Pull

Standard Operating Procedure (SOP)

http://www.onretrieval.com/images/PDF_s/Cellebrite_OnRetrieval_UFED_Touch_manual_usuario.pdf

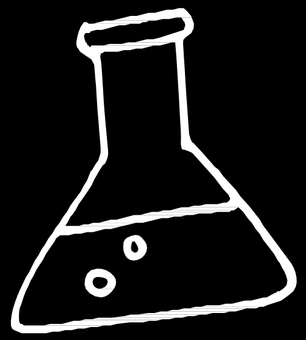
Mobile Forensic (physical/logical)
Analysis Tools

AUTOPSY
DIGITAL FORENSICS

MAGNET
FORENSICS®

Cellebrite

Android Forensic



Sample Mobile RAW Image

https://www.cfreds.nist.gov/mobile/index.html

Android Partition

/boot

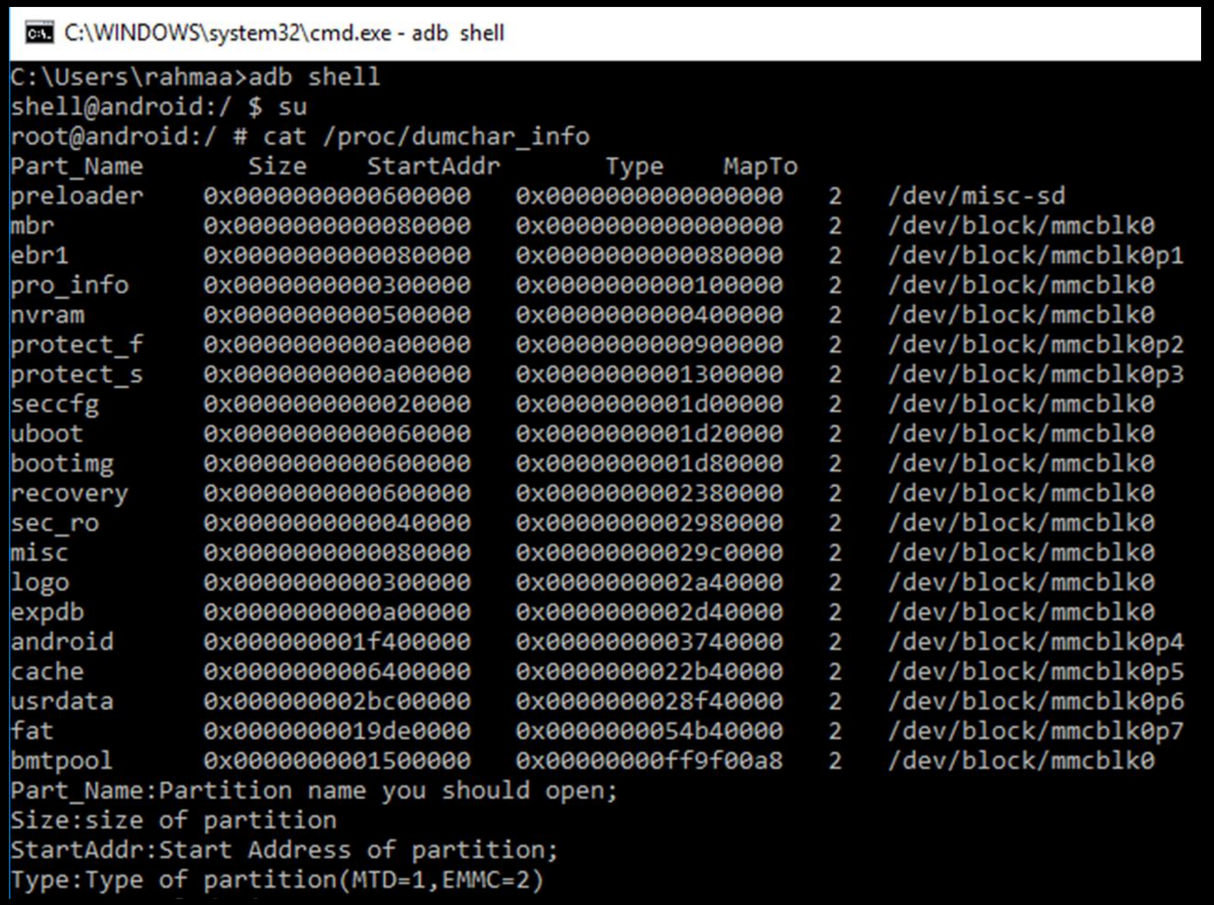
/cache

/usrdata

/misc

/recovery

/system

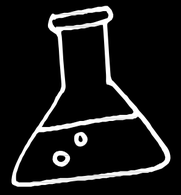


*All Android devices use separate partitions for storing different parts of the entire system

Glossary

https://source.android.com/devices/bootloader/partitions-images
http://onnocenter.or.id/wiki/index.php/ROM_Android:_Melihat_Partisi_ROM

Android Forensic



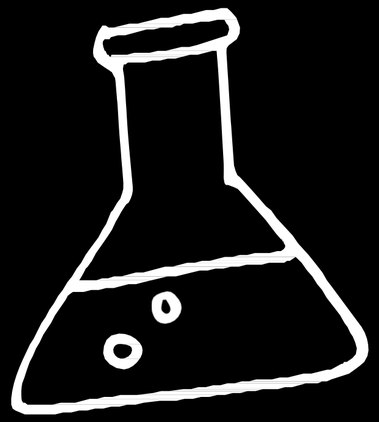
Android Artifacts

- ☑ /data/data/*package*
- ☑ /storage/sdcard0/Android/data/*package*

Glossary

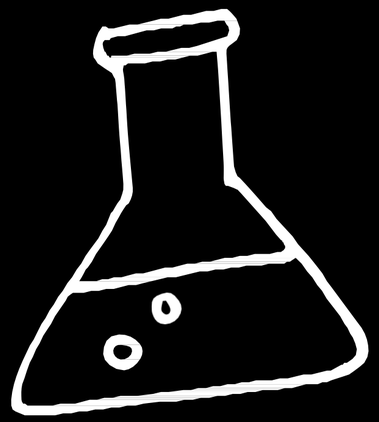
https://developer.android.com/training/data-storage/files.html#WriteExternalStorage
https://gist.github.com/lopspower/76421751b21594c69eb2

#Challenge 1 : FTK Imager, SQLite



- ☑ Whatsapp Messages
- ☑ SMS
- ☑ Contact

#Challenge 2 : Autopsy



- [x] File Image
- [x] Last Call Log
- [x] Last File Access
- [x] Email Address
- [x] Recovery Deleted Data
- [x] File Carving