OVERVIEW OF FORENSIC METODOLOGHY FOR APPROACHING FORENSICS

tes
texto.texto@proton.me
July 2024

AGENDA | 3 SESSIONS

Pre-assessment:

1. Core key questions
2. Contextual assessment, Vetting & Consent collection
3. Documentation
4. Preparation (lab, tools)

Assessment:

1. Data Acquisition
2. Forensics Analysis

Post-assessment:

1. Communication
2. Output of analysis (Report)
3. Follow-up (recommended actions and support)
4. Lessons learned

REFERENCES

Built on top of other organizations/individuals efforts:

Trainings:

- Digital Forensics Fellowship from Amnesty International
- Digital Defenders workshop on Forensics (Jacobo Najera & Marla)

Online resources:

- Guide to forensics Security Without borders | Garnieri & Etienne (https://github.com/securitywithoutborders)

Tools development:

- MVT project (https://github.com/mvt-project)

& personal perspective of field work

SESSION 1

Forensics from a Civic Society Organizations perspective

- consensual, respectful
- exploratory approach
- constrains on literature, tools & technical limitations

Core questions:

What is our aim? In what type of org is framed our work?
What is the expected outcome? What are our resources? What type of cases we will be receiving?

Initial triage

a. Contextual assessment. b. Vetting. c. Gather contextual facts

d. Explain the process. e. Consent collection. f. Risk analysis

DATA ACQUISITION

The process of extracting data from a device for analysis

3 types of acquisition methods:

a. Manual extraction: manually browsing the device via UI.
Ex. checking the permissions, content and installed apps.

b. Logical extraction: extract data from OS filesystem
Ex. recover application/user data using adb tool in Android (some available data or whole file system in rooted device).

c. Physical extraction: create a bit to bit copy of device
Ex. includes deleted/unallocated data. Constrains: file-level encryption in Android.

DATA ACQUISITION

Collecting device **artifacts**.

Artifact: piece of data to be analyzed, to understand if it is evidence relevant to our investigation (supports/refuses hypothesis).

- SMS links: check if malicious
- Browser history
- Network connections
- Programs / apps installed:
  - check if legitimate
  - capabilities/permissions
  - persistence
- Process running

DATA ACQUISITION

Collecting device **artifacts**.

- Files location & contents
- Insecure state of device:
  - non-secure configurations
  - disabled protections
  - signs of rooted devices
- Configuration of accounts
  - unknown sessions
  - trusted devices
  - tracking enabled
- Logs
- Time framed events / A combination of events
- ...

DATA ACQUISITION

Sources of data where to find those artifacts:

Android:

- Bugreport
- Via an adb connection
- Backup
- Via androidQF

iOS:

- Backup
- Sysdiagnose

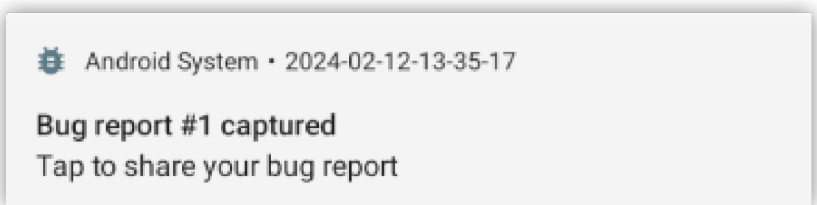
DATA ACQUISITION

ANDROID

1. BUGREPORT

- contains diagnostic information
  - info on system services (dumpsys)
  - error logs (dumpstate)
  - system logs (logcat)

- Easy to be generated
- Small size
- Not private data



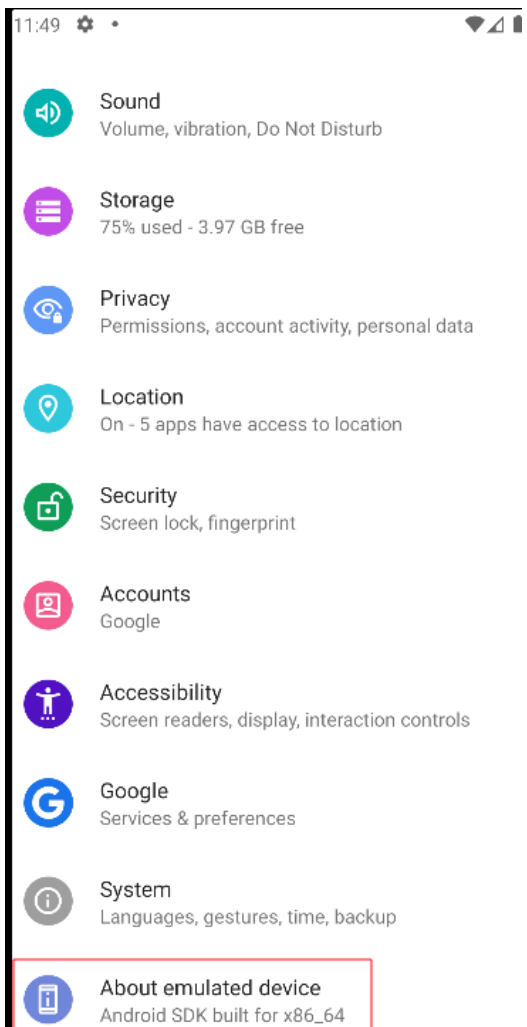
https://source.android.com/docs/core/tests/debug/read-bug-reports

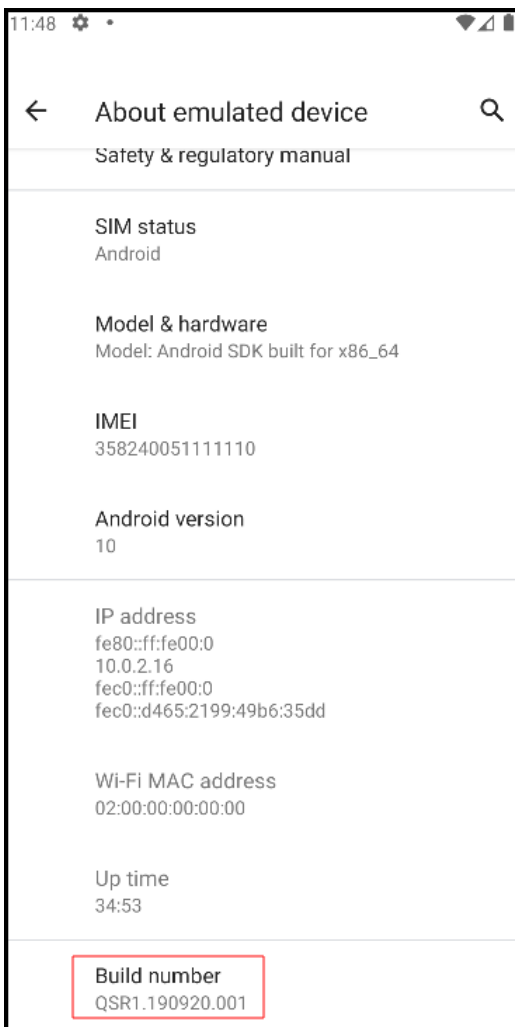
DATA ACQUISITION

ANDROID

1. BUGREPORT

a. How to:



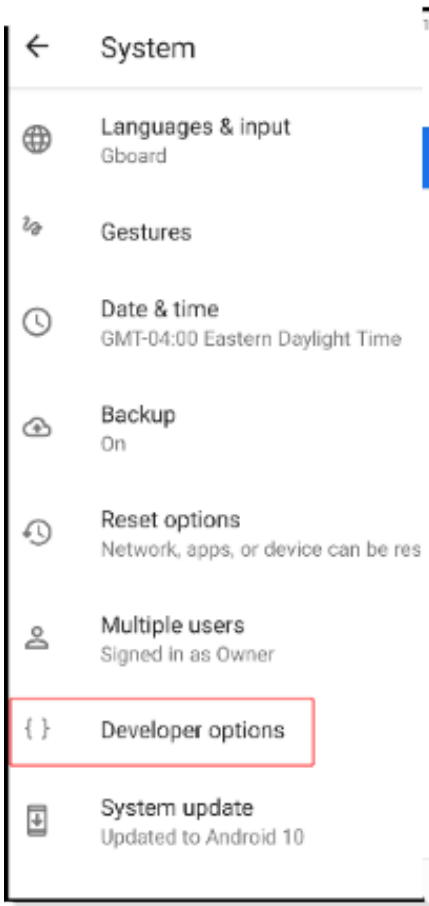


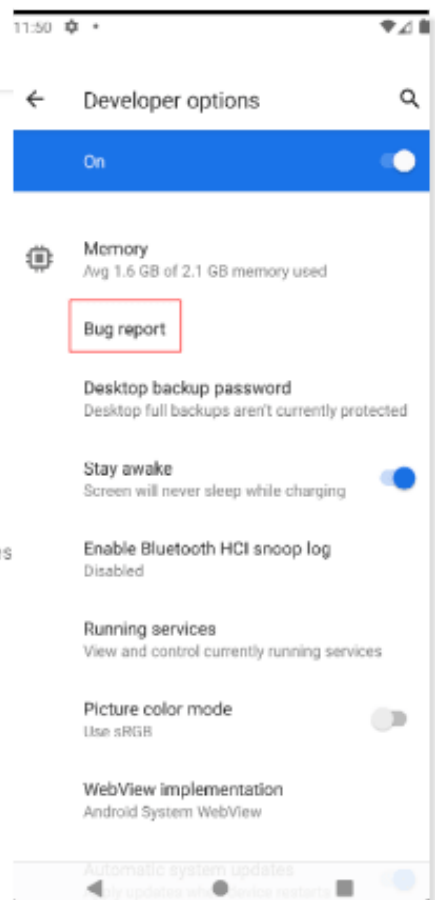
DATA ACQUISITION

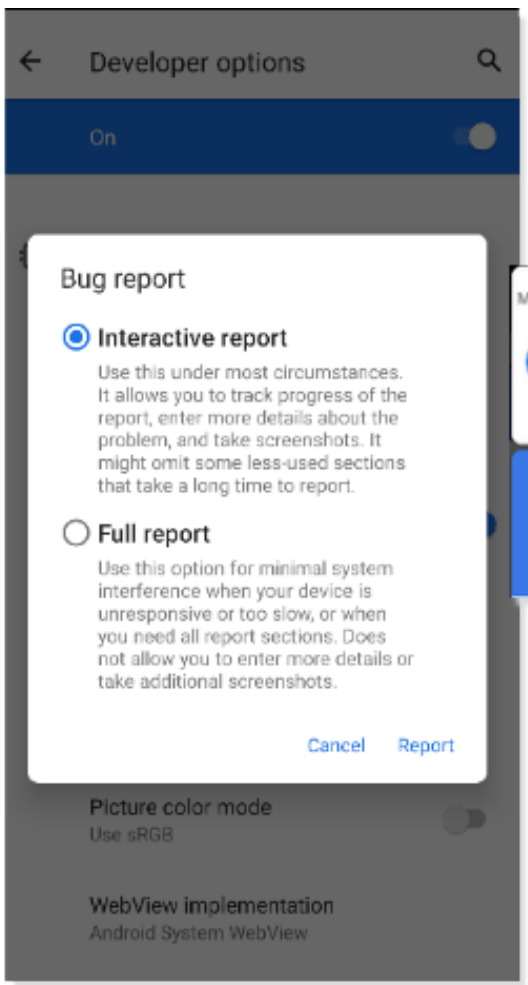
ANDROID

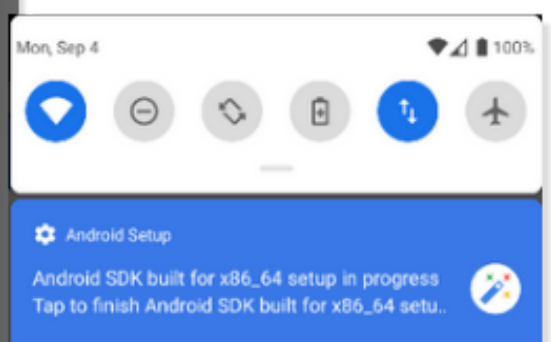
1. BUGREPORT

a. How to:









DATA ACQUISITION

ANDROID

1. BUGREPORT

b. Output files:

```
> ls -laR bugreport-java_retail-RTAS31.68-66-3-2024-02-12-13-35-17
bugreport-java_retail-RTAS31.68-66-3-2024-02-12-13-35-17:
total 40916
drwxr-xr-x 5                 4096 Feb 12 14:12 .
drwxr-xr-x 3                 4096 Feb 12 14:12 ..
-rw-rw-r-- 1             41791322 Feb 12 13:36 bugreport-java_retail-RTAS31.68-66-3-2024-02-12-13-35-17.txt
-rw-rw-r-- 1                42482 Feb 12 13:36 dumpstate_log.txt
drwxr-xr-x 5                 4096 Feb 12 14:12 FS
drwxr-xr-x 2                12288 Feb 12 14:12 lshal-debug
-rw-rw-r-- 1                   60 Feb 12 13:35 main_entry.txt
drwxr-xr-x 2                 4096 Feb 12 14:12 proto
-rw-rw-r-- 1                    3 Feb 12 13:35 version.txt
-rw-rw-r-- 1                21755 Feb 12 13:35 visible_windows.zip
```

DATA ACQUISITION

ANDROID

1. BUGREPORT

b. Output files:

```
> cat bugreport-java_retail-RTAS31.68-66-3-2024-02-12-13-35-17.txt | grep "DUMP OF SERVICE"
DUMP OF SERVICE CRITICAL SurfaceFlinger:
DUMP OF SERVICE CRITICAL activity:
DUMP OF SERVICE CRITICAL cpuinfo:
DUMP OF SERVICE CRITICAL input:
DUMP OF SERVICE CRITICAL notification:
DUMP OF SERVICE CRITICAL power:
DUMP OF SERVICE CRITICAL sensorservice:
DUMP OF SERVICE CRITICAL window:
```

```
> cat bugreport-java_retail-RTAS31.68-66-3-2024-02-12-13-35-17.txt | grep -A 30 "SYSTEM LOG"
------ SYSTEM LOG (logcat -v threadtime -v printable -v uid -d *:v) ------
--------- beginning of system
02-12 08:32:04.860  1000   883   922 I UsbPortManager: ClientCallback V1_1: port0
02-12 08:32:04.861  1000   883   905 I UsbPortManager: USB port changed: port=UsbPort{id=port0, sup
tPresenceProtection=falsesupportsEnableContaminantPresenceDetection=false, status=UsbPortStatus{con
supportedRoleCombinations=[sink:device], contaminantDetectionStatus=0, contaminantProtectionStatus=
onnectedAtMillis=22939, lastConnectDurationMillis=0
02-12 08:32:04.861  1000   883   922 I UsbPortManager: ClientCallback V1_1: port0
02-12 08:32:04.862  1000   883   883 I UsbDeviceManager: updateHostState UsbPort{id=port0, supporte
enceProtection=falsesupportsEnableContaminantPresenceDetection=false status=UsbPortStatus{connected
tedRoleCombinations=[sink:device], contaminantDetectionStatus=0, contaminantProtectionStatus=0}
02-12 08:32:04.862  1000   883  1208 I UsbPortManager: ClientCallback V1_1: port0
02-12 08:32:04.863  1000   883   922 I UsbPortManager: ClientCallback V1_1: port0
```

DATA ACQUISITION

ANDROID

2. VIA ADB

Android Debug Bridge (adb) is a command line tool that allows you to communicate with a device & gather a lot of information.

```
user@host:~

[user@host ~]$ adb devices
* daemon not running; starting now at tcp:5037
* daemon started successfully
List of devices attached
1e778e25        unauthorized

[user@host ~]$ adb devices
List of devices attached
1e778e25        device

[user@host ~]$ adb shell
OnePlus5:/ $ uname -a
Linux localhost 4.4.21-perf+ #1 SMP PREEMPT Fri May 26 16:25:14 CST 2017 aarch64
OnePlus5:/ $
```

DATA ACQUISITION

ANDROID

2. VIA ADB

a. How to:

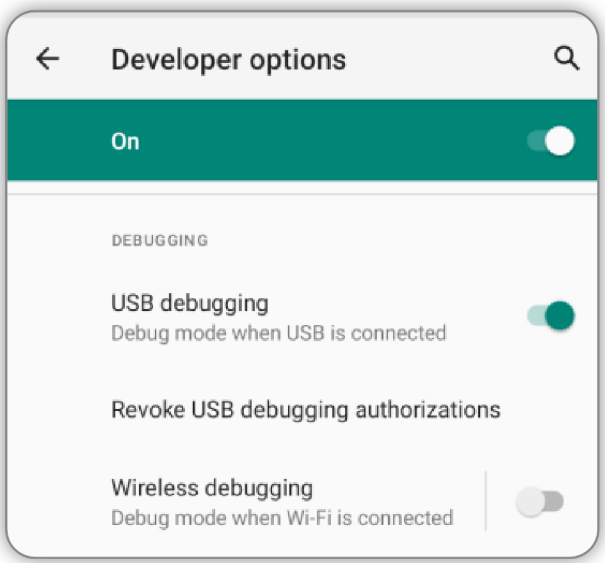
In the computer:

a. Install adb utilities
https://developer.android.com/tools/releases/platform-tools

b. Connect the phone to it via USB

In the phone:

c. Enable Developer options



DATA ACQUISITION

ANDROID

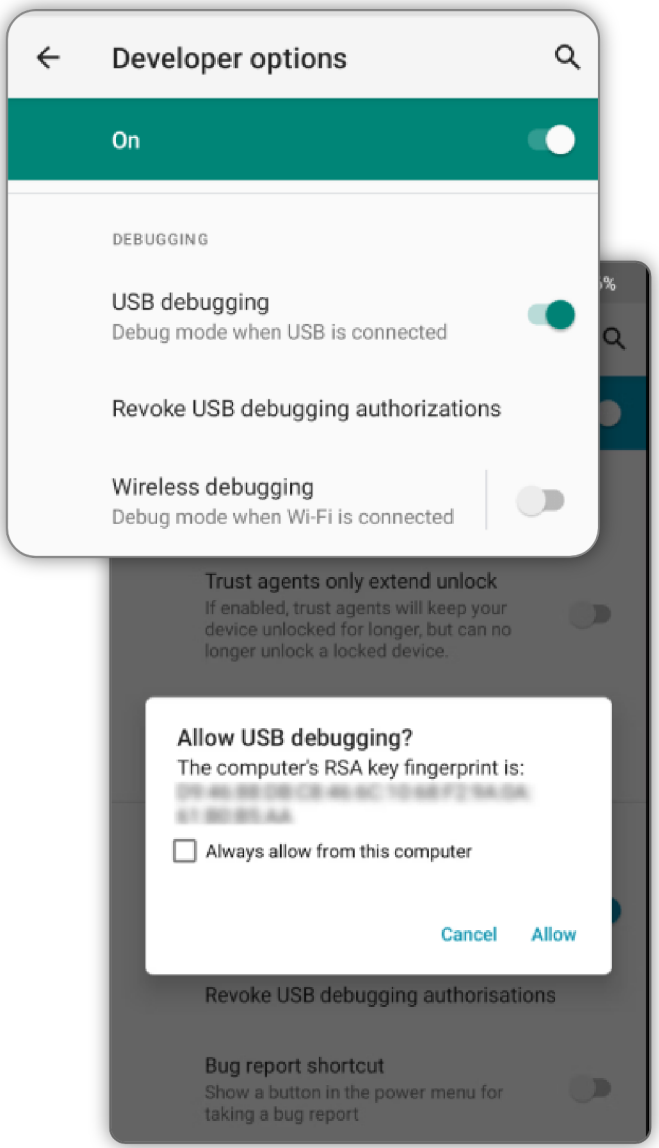
2. VIA ADB

a. How to:

In the phone:

d. Enable USB debugging: for the phone to communicate to computer

e. Authorize Host keys: once connected, the phone will pop-up a prompt for you to manually authorize the host keys



DATA ACQUISITION

ANDROID

2. VIA ADB

b. Output "shell"

In the computer:

```
> adb devices
List of devices attached
ZY32DBQN3T      device

> adb shell
java:/ $ ls
acct        config         default.prop     init.odm.carrier.rc      init.recovery.p354.rc
apex        d              dev              init.odm.rc              init.recovery.ums512_1h10.rc
bin         data           etc              init.recovery.common.rc  init.recovery.ums512_1h10_go.rc
bugreports  data_mirror    init             init.recovery.p352.rc    init.recovery.ums512_20c10.rc
cache       debug_ramdisk  init.environ.rc  init.recovery.p353.rc    init.recovery.ums512_2h10.rc
```

Be aware of type of access via adb is not privileged

DATA ACQUISITION

ANDROID

2. VIA ADB

b. Output "shell"

In the computer:

```
$ adb shell list packages
$ adb shell dumpsys [service]
$ adb shell service list
$ adb backup -apk -all -f backup.ab
$ adb bugreport
$ adb logcat
$ adb shell settings list [system|secure|global]

$ adb shell install <apk>
$ adb push <local> <remote>
```

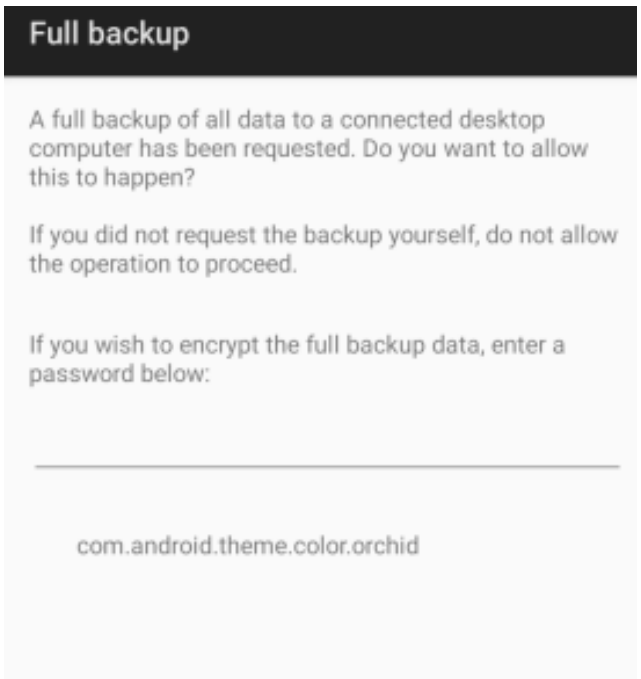
DATA ACQUISITION

ANDROID

3. ANDROID BACKUP FILE

- File with .ab format

-Contains:
  - backup data of apps with manifest flag `android:allowBackup=true` is being deprecated [not useful]
  - files in sdcard (ej. Pictures) [not useful]
  - SMS contents [useful]



DATA ACQUISITION

ANDROID

3. ANDROID BACKUP FILE

a. How to:

```
$ adb backup "com.android.providers.telephony"

# Use Backup extractor from https://github.com/nelenkov/android-backup-extractor
$ java -jar abe.jar unpack backup.ab backup.tar
$ 7z x backup.tar
```

Other (not needed):

```
$ adb backup -all
$ adb backup -all -shared
```

b. Output files:

- still zlib compressed

```
linux $ tree
.
└── backup
    ├── d_f
    │   ├── 000000_sms_backup
    │   ├── 000001_mms_backup
    │   ├── 000002_sms_backup
    │   ├── 000003_sms_backup
    │   ├── 000004_sms_backup
    │   └── 000005_sms_backup
    └── _manifest
```

DATA ACQUISITION

ANDROID

4. VIA ANDROIDQF

https://github.com/mvt-project/androidqf

- Eases the process of data acquisition
- Available for Windows, Linux and Mac.

```
> ./androidqf

                    __          _     __      ____
  ___  ___  ___/ /_______  (_)___/ /__   / __/
 / _ `/ _ \/ _  / ___/ _ \/ / _  / _ `/ /_
/ /_/ / / / / /_/ / /  / / /_/ / / /_/ / / /_/ / __/
\__,_/_/ /_/\__,_/_/   \____/_/\__,_/\__, /_/
                                      /_/
        androidqf - Android Quick Forensics

Started new acquisition a4c0e8aa-6a46-4108-832a-78858924a152
Would you like to take a backup of the device?
✔ Everything
Generating a backup with argument -all. Please check the device to authorize the backup...
Backup completed!
Collecting information on installed apps. This might take a while...
Found a total of 308 installed packages
Would you like to download copies of all apps or only non-system ones?
✔ Do not download any
Collecting device properties...
Collecting device diagnostic information. This might take a while...
Collecting list of running processes...
Collecting list of services...
Collecting list of files... This might take a while...
```

DATA ACQUISITION

ANDROID

4. VIA ANDROIDQF

a. How to:

- a. Enable developer Mode
- b. Connect via USB the phone to computer
- c. Accept pop-up
- d. Run androidQF

```
> ./androidqf

                                  __                _     __        ____
     ___   ___   ____/ /_______  (_)___/ /___   / __/
    / __ `/ __ \/ __  / ___/ __ \/ / __  / __ `/ /_
   / /_/ / / / / /_/ / /  / /_/ / / /_/ / /_/ / /_/ / __/
   \__,_/_/ /_/\__,_/_/   \____/_/\__,_/\__, /_/
                                          /_/
          androidqf - Android Quick Forensics

Started new acquisition a4c0e8aa-6a46-4108-832a-78858924a152
Would you like to take a backup of the device?
✔ Everything
Generating a backup with argument -all. Please check the device to authorize the backup...
Backup completed!
Collecting information on installed apps. This might take a while...
Found a total of 308 installed packages
Would you like to download copies of all apps or only non-system ones?
✔ Do not download any
Collecting device properties...
Collecting device diagnostic information. This might take a while...
Collecting list of running processes...
Collecting list of services...
Collecting list of files... This might take a while...
```

DATA ACQUISITION

ANDROID

b. Options during acquisition:

a. Saved folder with hash name

```
Started new acquisition afcd433d-567f-41b2-a2d4-8042b4cf7458
```

b. Backup

```
Would you like to take a backup of the device?
Use the arrow keys to navigate: ↓ ↑ → ←
? Backup:
    Only SMS
  ▸ Everything
    No backup
```

c. Download apps

```
Would you like to download copies of all apps or only non-system ones?
Use the arrow keys to navigate: ↓ ↑ → ←
? Download:
    All
  ▸ Only non-system packages
    Do not download any
```

```
Would you like to remove copies of apps signed with a trusted certificate to limit the size of the output folder?
Use the arrow keys to navigate: ↓ ↑ → ←
? Remove:
  ▸ Yes
    No
```

DATA ACQUISITION

ANDROID

4. VIA ANDROIDQF

c. Output files:

The result is a folder containing:

- Apps installed & its details in packages.json and ./apks
- A backup (full or only SMS)
- A partial list of files /sdcard & /tmp (files.json)
- System settings (settings*.txt)
- Info on receivers (dumpsys.txt)
- SELinux policy
- A summary (command.json)

```
> ls -la
total 133204
drwxr-xr-x 5          4096 Feb 12 16:39 .
drwxr-xr-x 9          4096 Feb 12 16:39 ..
-rw-r--r-- 1           479 Feb 12 16:39 acquisition.json
drwxr-xr-x 2         36864 Feb 12 16:37 apks
-rw-r----- 1      16768021 Feb 12 16:32 backup.ab
-rw-r--r-- 1        143746 Feb 12 16:39 command.log
-rw-r--r-- 1      15647113 Feb 12 16:38 dumpsys.txt
-rw-r--r-- 1          2883 Feb 12 16:39 env.txt
-rw-r--r-- 1      92226324 Feb 12 16:39 files.json
-rw-r--r-- 1         27487 Feb 12 16:37 getprop.txt
-rw-r--r-- 1         64590 Feb 12 16:39 hashes.csv
-rw-r--r-- 1      10560217 Feb 12 16:39 logcat.txt
drwxr-xr-x 5          4096 Feb 12 16:39 logs
-rw-r--r-- 1        625075 Feb 12 16:37 packages.json
-rw-r--r-- 1        205277 Feb 12 16:38 processes.txt
-rw-r--r-- 1             2 Feb 12 16:39 root_binaries.json
-rw-r--r-- 1             9 Feb 12 16:39 selinux.txt
-rw-r--r-- 1         10329 Feb 12 16:38 services.txt
-rw-r--r-- 1          7815 Feb 12 16:39 settings_global.txt
-rw-r--r-- 1          7913 Feb 12 16:39 settings_secure.txt
-rw-r--r-- 1          1278 Feb 12 16:39 settings_system.txt
```

Hands on!

DATA ACQUISITION

Sources of data where to find those artifacts:

Android:

- Bugreport
- Via an adb connection
- Backup
- Via androidQF

iOS:

- Backup
- Sysdiagnose

DATA ACQUISITION

iOS

1. ENCRYPTED ITUNES BACKUP

- A snapshot of the current state of the iPhone
- Contain a partial copy of the filesystem, including some of the user data and service databases with private information (SMS & call history, navigation history, Whatsapp history, calendar info, apps user info)
- System logs
- Encrypted backups provide more information

a. How to:

- Can be collected with a Mac or Windows through iTunes program https://docs.mvt.re/en/latest/ios/backup/itunes/

DATA ACQUISITION

iOS

1. ENCRYPTED ITUNES BACKUP

c. Output files:

In each backup, there are:

- Files with info:
  - Info.plist (info about device)
  - Manifest.mdbd (list of files in backup)
  - Manifest.plist (apps installed & info)
  - Status.plist (status of the backup itself)

Format: SQLite db, plaintext plist (XML like Property list files), binary plist and other non-standard.

Source: https://theapplewiki.com/wiki/ITunes_Backup

DATA ACQUISITION

iOS

1. ENCRYPTED ITUNES BACKUP

c. Output files:

- the files themselves with a new file name:

```
SHA1('HomeDomain-Library/SMS/sms.db') =
3d0d7e5fb2ce288813306e4d4636395e047a3d28
```

domain	path and file name	SHA-1 backup file name
HomeDomain	Library/SMS/sms.db	3d0d7e5fb2ce288813306e4d4636395e047a3d28
HomeDomain	Library/AddressBook/AddressBook.sqlitedb	31bb7ba8914766d4ba40d6dfb6113c8b614be442
HomeDomain	Library/Notes/notes.sqlite	ca3bc056d4da0bbf88b5fb3be254f3b7147e639c
WirelessDomain	Library/CallHistory/call_history.db	2b2b0084a1bc3a5ac8c27afdf14afb42c61a19ca

Source: https://theapplewiki.com/wiki/ITunes_Backup

DATA ACQUISITION

iOS

2. SYSDIAGNOSE

A diagnostic tool for troubleshooting and gathering system data. Collects logs, traces, and system state information. Gives a snapshot of the device's current condition.

- Running processes
- Key preferences files (plist)
- Network configuration & history
- Information on hardware health
- Log files
- Device diagnostic
- Usage overview

Reference:

https://www.first.org/resources/papers/conf2023/FIRSTCON23-TLPCLEAR-Durvaux-Using-Apple-Sysdiagnose-for-Forensics-and-Integrity-Check.pdf

DATA ACQUISITION

iOS

2. SYSDIAGNOSE

a. How to:

Needs to be generated in the device:

"a. Trigger a sysdiagnose by simultaneously pressing and releasing both volume buttons + the Side (or Top) button for 250 milliseconds. Holding too long (>1s) will lock the device instead. You will feel a short vibration.

b. Wait 10 minutes for the diagnostic gathering to complete.

c. Locate the sysdiagnose file:

Settings.app > Privacy > Analytics & Improvements > Analytics Data > “sysdiagnose_YYYY.MM.DD_HH-MM-SS-XX…tgz"

Source: https://podcasters.apple.com/assets/iOS-sysdiagnose-logging-instructions.pdf

DATA ACQUISITION

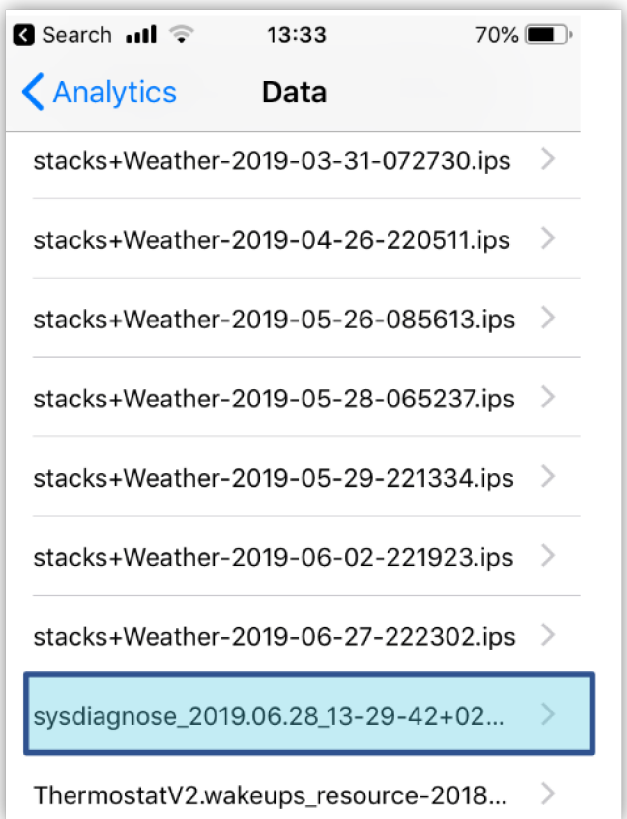
iOS

2. SYSDIAGNOSE

b. Output file:

The sysdiagnose file:

Settings.app > Privacy > Analytics & Improvements > Analytics Data > “sysdiagnose_YYYY.MM.DD_HH-MM-SS-XX...tgz"



Reference:

https://www.first.org/resources/papers/conf2023/FIRSTCON23-TLPCLEAR-Durvaux-Using-Apple-Sysdiagnose-for-Forensics-and-Integrity-Check.pdf

DATA ACQUISITION

How we do the acquisition?

- In person: lab machine with tools and space. Cables. Secure storing.

- Remote: tactic (instructions, remote desktop), time, good internet connection. Secure mean of sharing.

Once we have the data:

- No alteration (anti-tampering), loss, leakage.
- Preserve chain of custody. Use hashes.
- Do not work on the original.
- Be mindful we are being trusted with private personal information
- Leave device on same initial state (disable developer options, uninstall related software).

DATA ACQUISITION

Questions?