GitHub @deezombiedude622

VII. MOBILE FORENSICS

Mobile Forensics
- a branch of digital forensics
- focusing on data available in mobile devices (ex., smartphones, tablets, smartwatches, wearables)
- * targets info stored on device & in other remote location accessible fm device
  - ↳ incl. but ∅ limited to mobile device manufacturers' built-in & 3rd party online services

Supporting Drivers for Proliferation of Mobile Device Popularity
- * faster & cheaper connectivity
  - ↳ $ cost of data per MB ↓ over time
- * more online services
- * changes to "always-on"
  - ↳ mobile device "always on" compared to PCs' boot-up
- * consumption of multimedia (ex., images, video, music)
- * mobile penetration — # ppl possessing mobile devices ↑; # ppl possessing >1 mobile devices ↑
- * new business models (ex., ride hailing) thriving

Acquisition Types

Logical Acquisition
- examiner makes copy of files & folders in logical storage (ie., a partition), or those files\folders themselves
- (+) simplicity: logical objects easier to understand
- (+) less time-consuming
- (−) ∅ recover deleted items (may ∅ be aware that they exist)

File System Acquisition
- examiner makes copy of file sys's struct, incl. existing (logical) objs. lying on top of it
- (+) allows access to objs. (ex., SQLite db records) that were marked as deleted & ∅ available thru logical extraction
- (−) more time-consuming
- (−) slightly more complex;
  - ∴ requires better understanding of OS architecture & file sys's struct.

Physical Acquisition
- examiner makes bit-by-bit copy of storage media (on which lies file sys. & logical objs.)
  - ↳ v. similar to hard drive imaging in traditional computer forensic exam. of PCs
- (+) allows for identification & analysis of data remnants on actual storage media (ex., SD card)
- (−) requires direct access to flash memory inside device
  - ⇒ most complex way of data acquisition
  - * device manufacturers often secure hardware for direct access
    - ↳ vendors of mobile forensics tools need to employ various techniques to bypass restrictions
    - ex., upload own OS's kernel to device

Extraction Strategy Selection

Logical Extraction
- → Existing Data

Physical Extraction
- → Existing & Deleted Data (Automatic Decoding)
- → File\Data Carving (Manual Decoding)

Gaining Access to Mobile Devices

(Android) Rooting
- process that allows Android devices to overcome limitations set on equipment & software by manufacturers & distributors
- * user obtains "super-user" privileges + ability to change\replace sys. apps.

Permanent
- used in most cases; aimed at replacing sys. kernel to permanently take ctrl over device

Temporary
- mostly used by commercial mobile forensics software; usual device reboot will revert device to un-rooted state

(iOS) Jailbreaking
- allows full access to device & installation of apps\extensions ∅ normally available thru official distribution src, aka. Apple App Store

Tethered Jailbreak
- temporary jailbreak that resets after restart\shutdown; device ∅ work fully until jailbroken again (tethered via cable)

Semi-tethered
- temporary jailbreak that resets after restart\shutdown; device works as normal sans those requiring elevated privileges ∨ those outside normal operating params.

Untethered
- permanent jailbreak that survives reboot until sys. software upgrade

Restrictions on Rooting\Jailbreaking
- mobile device mgmt
- secure apps (ex., Internet banking, finance-related)
- device regarded as untrusted; feats ∅ functioning

Mitigations
- * enforcing detection & disable access to apps ∨ services
- * apply device policy to lock down & restrict any undesirable actions
- * implement in-app sandboxing (ex., V-key's V-OS)

Mobile forensics

Unique Principles

1. DON'T turn off powered-on devices
   - * access to various info will be lost
   - & device inaccessible after shutdown
2. Isolate network connection
   - * avoid remote wipe (a default feat.)
3. Proper identification
   - * unique identification for physical desc. & serial # required
   - ∴ device portability makes for easy mistaking & switcheroo
4. Post-mortem forensics
   - * "black-box" analysis based on what is available physically & digitally

Artifacts

- SIM-card related → mobile device's IMEI (International Mobile Equipment) nums.
  - ICCID (Integrated Circuit Card ID) num — SIM card # that links to id of user/owner of mobile phone
  - IMSI (International Mobile Subscriber Identity) num — refers to SIM card's country, mobile network, subscriber #
  - PIN & PUK nums, SIM apps.
- Online Accts.
  - pswd storage
  - browser history
  - phone manufacturer's default acct.
  - common apps (ex. O365, Dropbox, OneDrive)
- Phone Memory
  - location
  - calendar
  - contacts
  - multimedia (pics, video, audio)
  - SMS, MMS, msging apps
  - email
  - recent call list

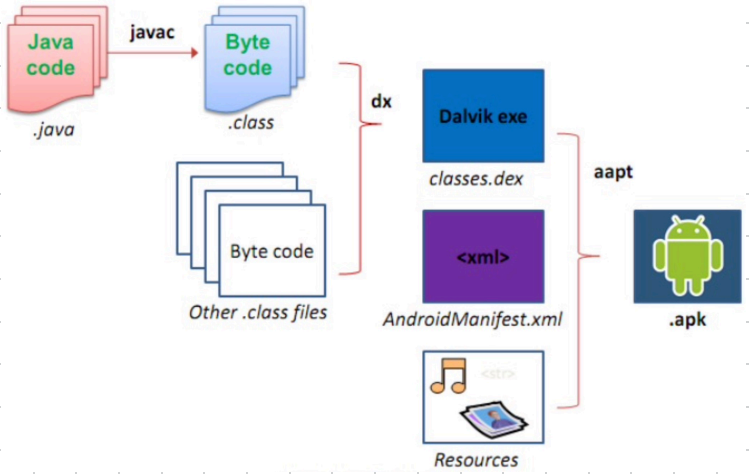
Specialized Mobile Forensics Tools

- → MobilEdit
- → Oxygen
- Cellebrite
- Axiom

linkedin.com/pulse/android-static-analysis-fundamentals-smali-code-introduction

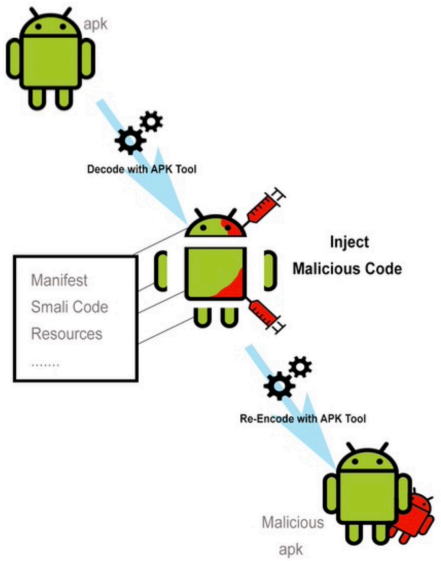
Note: smali code represents intermediate language of Android app bytecode
* bytecode can be converted into smali code
  - allows for examining an app's finality or behavior

Android App Package (APK)



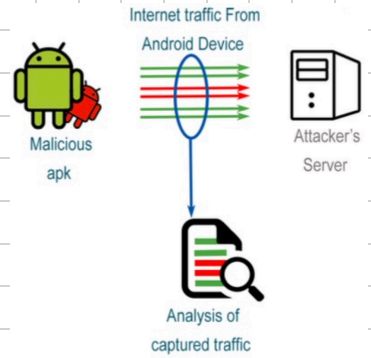
1. Prepare malicious payload, decode payload using APKtool.
   - * APKtool: used to reverse engineer APK to original form
2. Decode APK file using APKtool.
3. Edit mainactivity file (defines first activity \ first screen of app)
4. Add — payload smali files to original app file struct
   - new permissions to manifest file

Creation of Malicious APK



5. Re-encode APK file.

Aththanayaka & Ranaweera. (2022). Exploit an Android device using payload injected APK.



Installations using malicious APKs may invoke communication w/ atker's server.

Encryption in Mobile Devices

No Encryption (early mobile OS)
- ∅ encryption by default

Default Encryption (modern OS)
- employ encryption by default at boot

Default Encryption (Enhanced)
- employ encryption by default at file level
- * passing boot seq. ↛ unlock every file, just required sys. files only

Hardware Encryption data bits stored in flash chips encrypted at hardware level (AES grade)

Secure Chip & Enclave security chip to protect
- boot sequence
- firmware integrity
- biometric data

Secure Partitions software execution environment to implement simple mgmt & security services (ex., Samsung Knox)

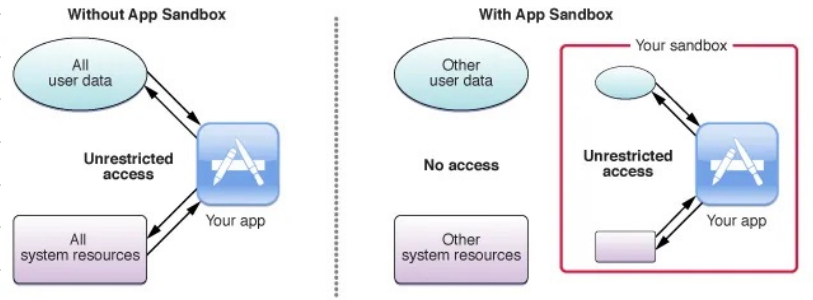
trusted firmware-a.readthedocs.io/en/v2.2/components/secure-partition-manager-design.html

App Sandboxing

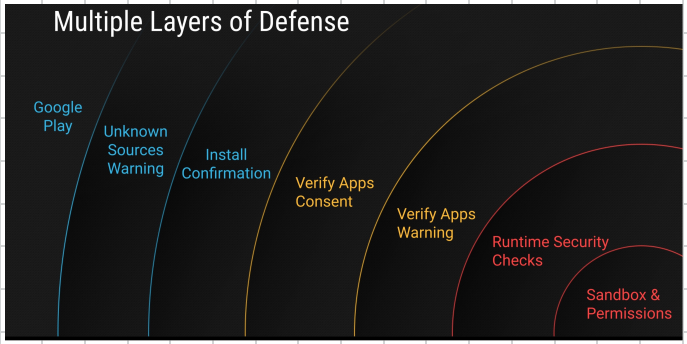
developer.apple.com/documentation/security/app_sandbox

source.android.com/docs/security/app-sandbox

- provides protection to sys. resources & user data
- by limiting apps access to resources requested thru entitlements (iOS) or user ID\UID (Android)
- * each app only has access to its own set of files or folders
- * sys. has access to all files & folders



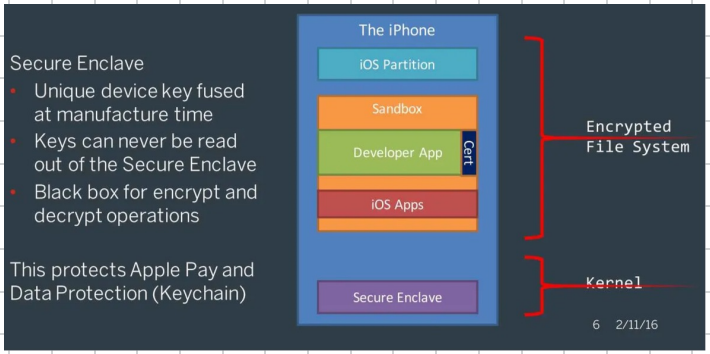
Android – Multiple Layers of Defense



Android Logical File System Structure

- ↳ /boot: files related to boot seq. (kernel, bootloader); required to boot
- ↳ /system: all OS files except kernel & bootloader; required to operate
- ↳ /recovery: backup of /boot & /system files to restore to default condition
- ↳ /data: user data files (user-created files)
- ↳ /cache: working files to speed up op. of certain apps
- ↳ /misc: other files (ex., settings, operator data, config data, USB, etc.)
- ↳ /sdcard: refers to built-in storage
- ↳ /sd-ext: refers to external\removable storage device

iOS Security Model



iOS Logical File System Struct.

HFS
- ↳ /dev/disk0s1s1
  - * mounted at root partition (/)
- ↳ /dev/disk0s1s2
  - * mounted at /private/var (apps, media files, settings)

AFS