iOS SANDBOXING
Lecture 19a

COMPSCI 702
Security for Smart-Devices

Nalin Asanka Gamagedara Arachchilage

Slides from Muhammad **Rizwan** Asghar

April 22, 2021

SANDBOXING

- A set of fine-grained access control that restricts each app to get access to its own resources

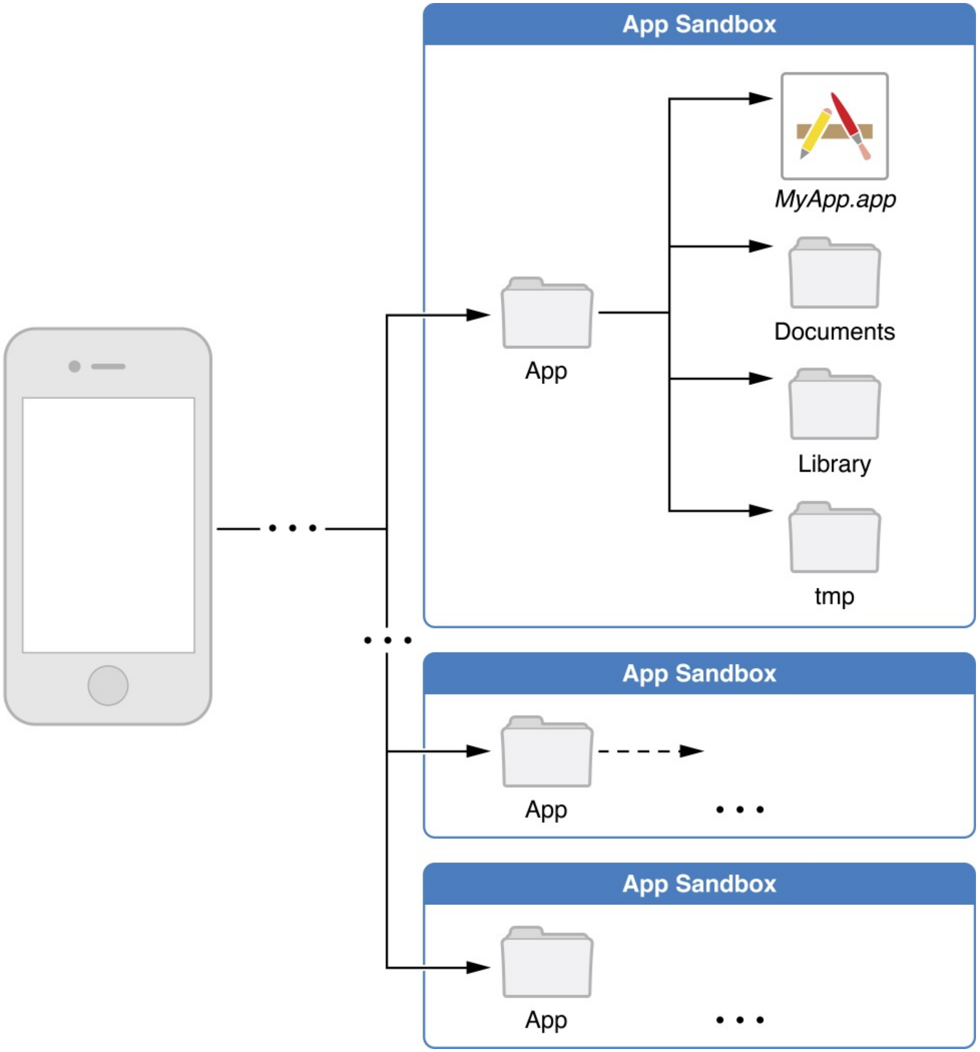
- Sandboxing limits what an app can do by maintaining a private environment of data for each app

- Sandboxing isolates app data and code execution from other apps

- The system installs each app in its own directory

iOS SANDBOXING

- When an app is installed on a mobile device, the system creates a unique folder for it



Source: http://www.cs.northwestern.edu/~ychen/classes/msit458-f13/BYOD_AlphaAlliance_part3.pptx

APP HOME DIRECTORY

Subdirectory	Description
<AppName>.app/	The signed bundle containing the application code and static data
Documents/	App-specific user-created data files that may be shared with the user’s desktop through iTunes’s “File Sharing” features
Library/	Application support files
Library/Preferences/	Application-specific preference files
Library/Caches/	App-specific data that should persist across successive launches of the application but not needed to be backed up
tmp/	Temporary files that do not need to persist across successive launches of the application

iOS SANDBOXING

- An app can read its own files but must get explicit permission for getting access to data of other apps

- Sandboxed apps store all the files, cookies, caches and other automatically generated contents in container directories

- A sandbox limits the damage that a potential hacker can do to an Apple iOS device

- Jailbreaking removes built-in sandbox restrictions

THIRD PARTY AND PLATFORM APPS

- All third party apps use the same profile but are each assigned their own container on the device filesystem
  - The container is stored in */var/mobile/Applications/UUID*
  - UUID is randomly generated at install (or re-install) time

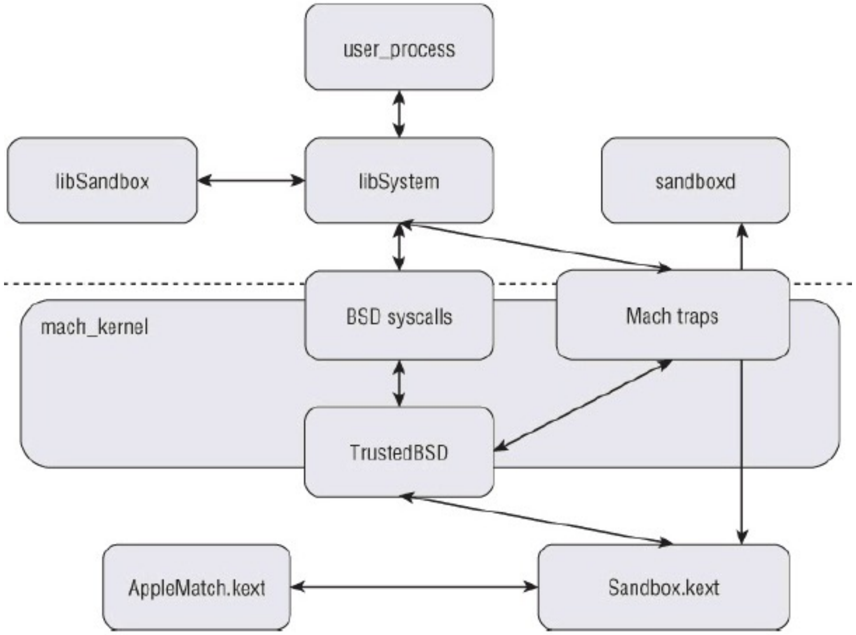
- Platform apps (built-in) have their own profiles
  - More than 40 platform apps have their custom profiles
  - E.g., the MobileSafari profile is only used by the MobileSafari application

MAC FRAMEWORK

- A sandbox is an access control system
- The sandbox is implemented using a policy module
  - User space configurable per process profile
  - Components
    - User space library functions for configuring and starting the sandbox
    - A kernel extension (with regular expression support) to evaluate policy restrictions
    - A kernel extension to enforce individual policies
    - A Mach server for handling logging

HOW DOES IT WORK?

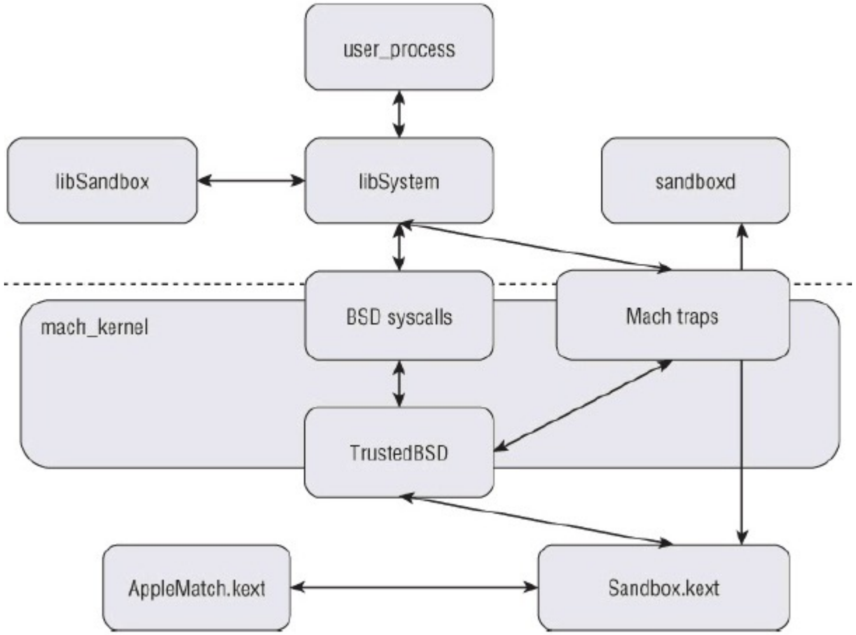
- On load (of an executable), sandboxing begins with a call to *sandbox_init*
  - A function of *libSystem*
- *sandbox_init* uses *libSandbox* to convert a human-readable policy into a binary format that the kernel expects
- The binary format is passed to *mac_syscall*
- It is handled by the TrustedBSD subsystem



Source: “iOS Hacker’s Handbook”

HOW DOES IT WORK?

- TrustedBSD passes the sandbox initialisation request to *Sandbox.kext*
  - A kernel extension
- The kernel extension installs the sandbox profile rules for the current process
- Upon completion, a return value is sent back



Source: “iOS Hacker’s Handbook”

SANDBOXING BENEFITS

- It protects app’s data by shielding it from other apps

- An app can freely store sensitive information in its own container

- It restricts apps to their designed function

- If the app is compromised (say through exploits), the attacker is limited to that container
  - It limits the damage malware can do to the device

SANDBOXING DOES NOT PREVENT MANY THINGS

- Apps are allowed to
  - Make network connections
  - Execute binaries from their application bundle directory
  - Send signals to themselves
  - Create sockets to receive kernel events
- Most built-in apps are not restricted
  - But MobileSafari and MobileMail do have their sandboxes
- Sandbox profiles can also limit memory and CPU cycles for an app

HUMAN READABLE POLICIES

- Only for non-default profiles
  - Default ones are already in a binary format
- Uses a domain specific language
- Sandbox Profile Language (SBPL)

```
(deny default)
(allow file-read-data
(literal “/var/whatever”))
```

- An ordered sequence of rules
- The first rule with a matching filter determines the result for the requested operation

SPYPHONE

- Developed by Seriot Nicolas (before iOS 6)
- Tested the sandbox
  - Could access
    - Cell phone number
    - Read/write access to address book
    - Safari and YouTube search terms
    - Email account info
    - Keyboard cache
    - Geo-tagged photos
    - GPS info
    - WiFi access point names
- Even inside a sandbox, a malicious app could extract a frightening amount of information from the device

SHARING DATA

- Since apps are constrained to their sandboxes, how do they share data?
  - Very limited channels
- Apps with the same *ApplicationIdentifierPrefix*
  - Which means the same developer
  - Can share data through the keychain
    - Originally just for passwords
    - But can take any data
- Can also share data via servers
- And of course via the clipboard (pasteboard)

SUMMARY

- Sandboxing isolates app data and code execution from other apps
- There are limited channels to share data under sandboxing environment

RESOURCES

- **iOS Hacker's Handbook**
  Charlie Miller, Dionysus Blazarkis, Dino Dai Zovi, Stefan Esser,Vincenzo Iozzo, Ralf-Philipp Weinmann
  John Wiley & Sons, Inc., 2012

- **Apple iOS 4 Security Evaluation**
  Dai Zovi, Dino A
  Black Hat USA 2011
  http://media.blackhat.com/bh-us-11/DaiZovi/BH_US_11_DaiZovi_iOS_Security_WP.pdf

- **App Sandboxing**
  https://developer.apple.com/app-sandboxing/

RESOURCES (2)

- **Sandbox in Depth**
  https://developer.apple.com/library/prerelease/mac/documentation/Security/Conceptual/AppSandboxDesignGuide/AppSandboxInDepth/AppSandboxInDepth.html

- **XiOS: Extended Application Sandboxing on iOS**
  Bucicoiu, Mihai, Lucas Davi, Razvan Deaconescu, and Ahmad-Reza Sadeghi
  In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, pp. 43-54. ACM, 2015
  https://www.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_TRUST/PubsPDF/XiOS.pdf

ACKNOWLEDGEMENT

- Some of the slides are based on the presentation shared by Robert Sheehan, thanks to him!

Questions?

Thanks for your attention!