Mobile App Taxonomy

Native Apps

- Developed for iOS on Mac in Xcode
  - Written in Objective-C or Swift
- Developed for Android on any device in Android Studio SDK
  - Written in Java or Kotlin
  - NDK available for system libraries in C/C++
- Access to all device functions and OS APIs
- Smoother and faster UX
- Platform dependent – requires re-writing app for other platforms

Web Apps

- Written in HTML 5/CSS/Javascript
- Runs in a mobile browser
- Primarily a mobile interface to a website
- Can be packaged as an app, with an icon for better user experience
  - Not accepted in Apple App Store
- Security issues similar to a standard website application
- Advantages:
  - Easy to develop, especially for web developers
  - Cross-platform
- Disadvantages:
  - Less access to sensors, though HTML 5 does provide APIs
  - Slower, less attractive UX

Progressive Web Apps

- Invented by Apple for early iPhones
  - Was ignored for years, but is becoming very popular
- Much like a Web App, but content is downloaded from browser and available when offline
- Can be used for large apps to install main app, and download content later for offline use
  - This sideloading process also opens up the opportunity to load potentially malicious code or content, not approved by the app store
- Android also has a similar concept called Instant Apps

Hybrid Apps

- Offer ability to develop one code base for multiple platforms
- A web-to-native abstraction layer or virtual machine converts web technology code to use more native API's
- Some run in a webview, an app that is similar to a browser
- Advantages
  - Same as Web Apps, but with better hardware support
- Disadvantages
  - Can be slow because of abstraction layer
  - Some types open security issues

Hybrid App Toolkits

Development tools used to port a single code base to two platforms

- Generally require their own specific development syntax
- Apache Cordova – open source fork of Adobe PhoneGap
- Ionic- Web technology apps similar to Cordova
- Flutter - Google
- React Native - Meta
- Unity
- Xamarin - Microsoft

Identifying Hybrid Apps

Unity Framework

- Unity is a popular platform for developing 2D and 3D games, but is also used for some utility apps
- Apps can be coded in C#, JavaScript or Boo (a language for .net)
- Unity engine converts apps to native iOS or Android code at run time
- Distributed as an IPA for iOS or an apk for Android

Unity Framework App Structure

- Android apk uses standard structure but `classes.dex` only contains a stub of code to call the Unity engine
- Binaries are stored as .dll files in assets/bin/Data/Managed
- Assets stored in assets/bin/Data
- Can be analyzed with strings command and dotPeek .net decompiler - https://www.jetbrains.com/decompiler/

Xamarin Framework

- Microsoft owned cross-platform development framework
- Apps are coded using C#, and then generate runtime versions for multiple platforms.
- Xamarin apps are distributed as CIL files with a DLL extension. The CIL files are interpreted at runtime and converted to native machine instructions using the Mono framework
- DLL files can be extracted in Windows

Xamarin Framework App Structure

- Xamarin Android apps include a NOTICE file in the top of the archive

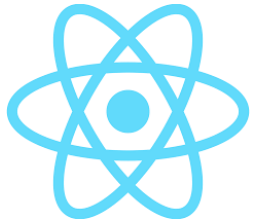
```
$ cat NOTICE
Xamarin-built applications contain open source software
For detailed attribution and licensing notices, please visit:
http://xamarin.com/mobile-licensing
```

- Xamarin apps distribute the DLL files in the assemblies directory, with the Mono runtime libraries distributed in the lib directory
- Can be analyzed with strings command and JustDecomiple - https://www.jetbrains.com/decompiler/

Flutter Framework

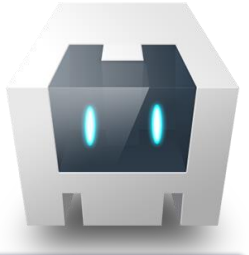
- Apps written in Dart programming language and built with widgets
- Large initial app size due to flutter engine
- Apps compile to native ARM code and have native-like performance
- Can be identified by searching apk for “flutter” often a flutter-assets folder

React Native



- Meta framework that might be familiar for web developers
- Uses Javascript and imported libraries
- Small app size but performance is slow on mobile due to importing libraries and running VM.
- Decompile code with jadx or another tool and search for com.facebook.react

Apache Cordova Framework



- Apps built in HTML5, CSS and JavaScript, making it attractive to web developers
- Apps can be exported into Android and iOS, and lots of other platforms you don't care about – Windows Phone, Tizen, FireFox OS, BlackBerry
- Use is fading in favor of other frameworks

Apache Cordova Framework App Structure

- File structure is similar to all other apks, but includes assets folder
- App functions are all JavaScript code in assets/www/js
- Cordova does NOT include obfuscation, so original source is often available to testers, including comments.
  - Third party obfuscation tools are available
    - https://jscrambler.com/
    - Thicket obfuscator for JavaScript
- Manipulating Cordova apps is trivial
  - Extract with apktool
  - Add, delete or edit any JavaScript Code
  - Build with apktool

Can Cordova Apps be Secure?

- Cordova makes it especially easy to recover api keys and other secrets
- It is easy to tamper with Cordova apps by modifying JavaScript
- Cordova should not be used for apps that require client-side enforcement
  - Server-side filtering should be included
- Developers should obfuscate the JavaScript
- Cordova is best suited for apps that do not have sensitive functionality

Mobile App Taxonomy Summary

- Types of apps
- Advantages and Disadvantages
- Security concerns
- Details for identifying hybrid apps