# Mobile App Testing Tools

# Assignment 1 Debrief

- Intended to be a scavenger hunt
  - See what you can find to get familiar with report and vulnerabilities
- iOS
  - Webview
  - Insecure random numbers
  - Camera usage
  - Logging –local data storage
- Android
  - Insecure broadcast receiver
  - Weak crypt signatures and Janus
  - Hardcoded secrets

# Automated App Testing

# Automated App Testing

- Rapidly evolving tools available that allow creation of automated scripts to remotely run regression test cases on specific devices and operating systems
- Advantages:
  - Doesn't require a testing setup
  - Doesn't require testing skills
  - Fast
- Disadvantages:
  - False positives
  - Trivial for a malicious app to identify the presence of an analysis environment versus a typical mobile environment
  - May not meet many testing criteria

# MITRE App Vetting Tools Analysis

Stoplight chart comparing the criteria satisfied, partially satisfied, and not satisfied by the Android tools.




*Source: https://www.mitre.org/sites/default/files/publications/pr-16-4772-analyzing-effectiveness-mobile-app-vetting-tools-report.pdf*

# Free App Testing Resources

- Use as a comparison or guide for your own tests
- Free tests against OWASP Top 10
  - https://www.immuniweb.com/mobile/
- One free test
  - https://www.ostorlab.co/
- One free test
  - https://quixxisecurity.com/pricing/

# Open Source Automated Static Testing

- Safe to run on questionable apps, because code is not executed
- Some open source tools available
- Quick Android Review Kit - https://github.com/linkedin/qark/
  - No root required
  - Designed to look for several security related Android application vulnerabilities, either in source code or packaged APKs
  - If vulnerabilities are found, it can generate an exploit apk
- Mobile Security Framework (MobSF)
  - https://github.com/MobSF/Mobile-Security-Framework-MobSF
  - Static and Dynamic testing

# Install Local MobSF

- Install Docker from repository
  - https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository
- Install Mobile Security Framework Docker
  - https://mobsf.github.io/docs/#/mobsf_docker

# App Testing Toolkit

# General App Testing Toolkit

- Device or emulator to run app
  - Rooted or Jailbroken is preferred
  - Don't use your primary device
- Android Studio for Android
- Xcode for iOS
- Burpsuite or other proxy
- Wireshark

# Overview of App Testing Devices

- Simulators/Emulators
  - Interaction with the actual device hardware features such as a camera or accelerometer cannot be simulated and requires an actual device.
- Remote Device Access
  - Allow the analyzer to view and access an actual device from a computer. This allows the testing of most device features that do not require physical movement.

# Testing on a Real Device

- Usually, must be rooted or jailbroken for some processes
  - Downside is this will also trigger apps that detect rooted devices
- Requires network setup for communication testing
  - Connect to same wireless AP as analysis computer or ad hoc connection directly to computer
- Requires a proxy on the analysis computer to intercept traffic or dumping traffic from device
  - Requires configuring proxy certificate to be trusted on device
  - May require decompiling the app and updating it's trusted certificates

# Testing on an Emulator

- May not be able to test an app properly in an emulator if the app relies on a specific mobile network or uses NFC or Bluetooth
- Possible to emulate many hardware characteristics, such as GPS and SMS
- Still may require proxy and certificate configuration
- Android emulators
  - Android Virtual Device in SDK – current best solution
  - Genymotion
  - Nox
  - Corellium – commercial
    - for Android and iOS

# Testing on an Emulator

- Pros:
  - Less expensive
  - Easier to restore, take snapshots, reset
  - Can choose API level and use different ones
  - Can be rooted
  - Uses true system libraries
- Cons:
  - More difficult user interaction
  - Poor app performance
  - Uses true system libraries

# Testing in the Cloud

- https://www.browserstack.com/app-live

# Testing Setup for Android

- Can be done on Windows, Linux or Mac
- Android Studio
  - SDK and platform tools, like emulator
- Can test on a real, rooted device or emulator
  - Magisk for rooting
- APK Extractor
- Objection/Frida
- ADB
- Burpsuite
- Wireshark
- Jadx (and possibly Ghidra) for reverse engineering
- SQLite for database investigation

# Sources for apk Files

- Google Play Store whenever possible
  - Extract from device with adb
  - Usually requires renaming
- APKMirror
- APKPure

# Testing Setup for iOS

- Mobile Security Framework for basic overview
- Mac computer
- Xcode
- Jailbroken device
  - Xcode offers “simulator”, for testing app functions, but not good for security
  - Corellium offers commercial emulator
- Cydia to install IPA
- iFunBox for file management on phone
- Frida
- SQLite for database investigation

# App Vetting for this Course

- We will experiment with vulnerable apps and several tools in class
- You will adapt what we do in class to investigate an app of your own choosing and provide a report at the end of the course
- **Only download potentially malicious apps directly to emulator or virtual machine**
  - Some are known to contain host viruses

# Assignment 1: Setup Testing Tools

- Setup and test Android Studio
- MobSF.live can be slow and glitchy depending on current activity
- To install your own version, setup a linux VM and use instructions in previous slide

# Summary

- Overview of app vetting tools
- Automated and web-based tools
- Testing on an emulator vs real device
- App testing tools