Defending Mobile Applications

Presented by Jerod Brennen, CISSP

CTO & Principal Security Consultant, Jacadis
Overview

- Layered Security
- iOS vs. Android
- An Attacker’s Perspective
- Best Practices
- Resources
Layered Security
Philosophy

"A Jedi uses to force for knowledge and defense...never for attack."

- Master Yoda

“It is said that if you know your enemies and know yourself, you will not be imperiled in a hundred battles.”

- Also Master Yoda
  (or maybe Sun Tzu)
Layered Security

- Four (4) Key Areas
  - Application
  - Sandbox
  - Operating System
  - Network

- Three (3) Types of Mobile Apps
  - Web (browser-based)
  - Native
  - Wrapper

- Document the data flow
  - Data at rest
  - Data in motion
  - Integration points
iOS vs. Android
Application Security Models

iOS
- Sandboxing (One Folder per App)
- Files
- Preferences
- Network Resources

Android
- Sandboxing (One ID per Package)
- Apps start with no permissions
  - `<uses-permission>` tags in `AndroidManifest.xml` (for protected features)
  - Declare and enforce permissions with `<permission>` tags
App Stores

- **iTunes**
  - Annual developer fee ($99)
  - Developer must provide Apple with your SSN
  - Apps must be approved by Apple

- **Google Play**
  - $25 fee per app you submit
  - Developer must provide email, website, and phone number
  - No one ever abuses this system  
    *wink, wink, nudge, nudge*
  - Apps do not require Google’s approval

- **Amazon App Store**
  - Annual developer fee ($99)
  - First year waived
  - Developer must provide name, email, mailing address, and phone number
  - Apps must be approved by Amazon

- **Cydia (also, HackStore)**
  - App store installed on jailbroken iPhones / iPads / iPods
  - Cydia = package manager (installs apps from repos)
  - App security is linked to repo trustworthiness
An Attacker’s Perspective
Mobile Malware

TYPES OF MALWARE TARGETING MOBILE DEVICES

- 36.43% SMS Trojan
- 63.39% Spyware
- 0.09% SMS Flooder
- 0.09% Worm

Image from ‘Juniper Networks 2011 Mobile Threats Report’
Online Attacks

- **Rogue Apps**
  - Andoid.Pjapps (hijacked Steamy Windows app, sent premium texts)
  - DroidDream (sent user info to remote server)
  - DroidKungFu (installed back door, complete device compromise)
  - Plankton (Angry Birds ‘supplementary’ program, installed other files post-install)

- **Man in the Middle**
  - Mallory Proxy (Intrepidus Group)
  - Rogue Wireless AP

- **Social Engineering**
  - Phishing
  - Smishing
  - Serve malicious content via in-app ads
Offline Attack – App Deconstruction

- Download from app store
  - C:\Users\<username>\Music\iTunes\iTunes Media\Mobile Applications
- Extract app to folder using 7-zip
- Manually examine the files using Notepad++ or prgrep
- Look for sensitive info (integration points)
  - Connection strings
  - Calls to Internet-facing web services
  - Calls to other local resources
Offline Attack – iOS Forensics (Windows)

- Stand up a Windows 7 virtual machine
- Install iTunes
- Connect the device to the vm and backup via iTunes
- Open backup files with SQLite3Explorer
  - C:\Users\<User_ID>\AppData\Roaming\Apple Computer\MobileSync\Backup\n  - SMS: 3d0d7e5fb2ce288813306e4d4636395e047a3d28
  - Call Log: 2b2b0084a1bc3a5ac8c27afdf14afb42c61a19ca
- Use Query Builder to extract evidence
- Alternately, manually examine the files using Notepad++ or prgrep
Offline Attack – iOS Forensics (*nix)

- **iPhone Backup Analyzer (iPBA)**
- Stand up an Ubuntu virtual machine
  - I’m a fan of BackTrack 5 (release candidate 2)
- Download .tar of iPBA
- Command Line: `easy_install pyttk`
- Copy iPhone backup folder to the vm
- Run the iPBA Python script
  - `.main.py -d <backup_directory>`
- Examine additional info
  - Call history, thumbnails, contacts, network ID’s, SMS data, cell location data, safari history, notes
Best Practices
viaForensics

- Digital forensics and security firm
- Multiple projects
  - appWatchdog
  - AFLogical
  - Santoku Linux
- 42+ Secure Mobile Development Best Practices
viaForensics Best Practices – General

- Avoid use of query string for sensitive data
- Institute local session timeout
- Implement code complexity and obfuscations
- Use address space layout randomization
- Avoid simple logic (if sessionIsTrusted == 1)
- Avoid simple logic variables (session.trusted = TRUE)
- Test third party libraries
- Use geolocation carefully
- Limitations of UUID (Universally Unique Identifier)
- Avoid use of MEID (Mobile Equipment Identifier) as user identifier
- Tamper checking
- Implement enhanced / two-factor authentication
- Protect application settings
- Validate input from client

From https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/
viaForensics Best Practices – Server

- Web server: check session settings
- Prevent framing and clickjacking
- Web server configuration
- Protect against CSRF with form tokens
- Protect and pen test web services
- Protect internal resources
- Fully validate SSL/TLS
- Protect against SSLStrip
- Certificate pinning
- SSL configuration

From https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/
viaForensics Best Practices – Data Storage

- Avoid storing sensitive data on the device
- Avoid caching app data on the device
- Limit caching of username
- Avoid crash logs
- Disable debug logs
- Hide account numbers and use tokens
- Use SECURE setting for cookies
- Implement secure data storage
- Be aware of copy/paste
- Be aware of the keyboard cache
- Securely store sensitive data in RAM
- Understand secure deletion of data

From https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/
viaForensics Best Practices – Android

• Implement file permissions carefully
• Implement *Intents* carefully
• Check activities
• Implement *Broadcasts* carefully
• Implement *PendingIntents* carefully
• Validate *Services*
• Avoid *Intent* sniffing
• Implement *ContentProviders* carefully
• WebView best practices
• Avoid storing cached camera images
• Avoid GUI objects caching

*From https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/*
viaForensics Best Practices – iOS

• Avoid cached application snapshots
• Use the Keychain carefully

From https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/
Resources

- What, me worry?
  - Pen Testing Mobile Applications
  - OWASP Top Ten Mobile Risks
    - [http://www.slideshare.net/JackMannino/owasp-top-10-mobile-risks](http://www.slideshare.net/JackMannino/owasp-top-10-mobile-risks)

- Comparing Android to iOS
  - Android vs. iOS: Security Comparison
  - Android Application Security and Permissions
  - iOS Application Security
    - [http://www.dummies.com/how-to/content/application-security-on-apple-ios-mobile-devices.html](http://www.dummies.com/how-to/content/application-security-on-apple-ios-mobile-devices.html)
  - App Store Comparison
    - [http://bjango.com/articles/appstores/](http://bjango.com/articles/appstores/)
More Resources

• Mobile Antimalware
  • 10 Examples of Mobile Malware
    • http://www.boston.com/business/technology/gallery/smartphone_malware_examples/
  • Lookout
    • https://www.mylookout.com/
  • SMobile
    • http://www.smobilesystems.com/

• Mobile Security Reports & Recs
  • Juniper Networks 2011 Mobile Threats Report
  • NSA Mobility Capability Package
    • http://www.nsa.gov/ia/_files/Mobility_Capability_Pkg_(Version_1.1U).pdf
  • viaForensics 42+ Best Practices [for mobile app development]
Tools

- 7-Zip
  - http://www.7-zip.org/

- BackTrack Linux
  - http://www.backtrack-linux.org/

- iPhone Backup Analyzer
  - http://www.ipbackupanalyzer.com/

- Mallory: Transparent TCP and UDP Proxy
  - http://intrepidusgroup.com/insight/mallory/

- Notepad++
  - http://notepad-plus-plus.org/

- PRGrep
  - http://www.prgrep.com/

- Santoku Linux
  - https://santoku-linux.com/

- Sqlite3Explorer
  - http://www.singular.gr/sqlite/

- viaForensics App Watchdog
  - https://viaforensics.com/appwatchdog/
Questions?

Jerod Brennen
http://www.linkedin.com/in/slandail
https://twitter.com/#!/slandail

Jacadis
https://www.jacadis.com/
info@jacadis.com