Mobile Device Security

Pen testing and Ethical Hacking

Lee Neely, CISSP, CISA, CISM, CRISC, CCUV, GMOB

Overview

- Mobile Devices are prevalent
- Security models are different by platform
- Analysis of Mobile Device Applications
- Analysis of Mobile Device Ecosystem

Mobile Devices are everywhere

- Resistance is futile! If management blocks them, users will find a way
  - Don’t be the problem your users solve
- It is important to understand how they are being used
  - Where is the data?
- It is important to understand the major device architecture and security
  - How is your information accessible?
  - How can your protections be circumvented?
- Based on that knowledge, coupled with your information protection requirements, you can start to assess.

Mobile Device Security

- IOS Security
- Hardware Encrypted with passcode
- Up to 4 year device support
- No MO value added software. OS updates directly from Apple
- Applications only installed iTunes App store
- Applications announce optional security settings

- Android Security
- Hardware Encryption inconsistent
- Devices supported 6-12 months.
  - Handset alliance suggest 18 months
- MO value add and testing delay OS updates
- By Default Applications installed from Google Play only, can add multiple sources.
- Applications announce security settings
  - All or nothing

Analysis of Mobile Devices

- Need to understand network connections and security
- Illegal to intercept/manipulate 3G/4G/LTE networks
- Must assume Wi-Fi network connections/behavior identical to Cellular network
- Need to be MiTM with the mobile device
  - Configure Proxy
  - ARP Posoining
  - AVD Emulator

Application Analysis

- IOS Applications
- Written in Objective C
- Encrypted
- Multiple hardware architecture in single application binary
- Need GDB to extract binary from memory
- Very difficult to change and repackage without source.

- Android Applications
- Written in Java
- .APK file is compressed (Zipped) Java
  - Runs in Dalvik interpreter
- Can Decompile into Java for analysis
- Can Decompile into Smali (Dalvik bytecode) for alteration
- Altered code can be assembled and distributed as the original package

iOS Decompilation/Manipulation

- Obtain unencrypted binary
  - Use Otool and GDB on JB iPhone
  - Use Rasticrac on JB iPhone
- Thin application to current architecture
  - Use lipo on OSX with Xcode installed to extract desired binary. E.g. ARMv7
  - Use class-dump-z to extract class files, headers pretty accurate
  - Otool for review of shared libraries
- Cannot recreate (modified) app

- Runtime Manipulation
- Connect to running app with Cycript
  - Cycript sits on Cydia Substrate
  - Manipulate application in memory
  - Dynamically access methods and variables
  - Is a testing/developer tool

Android Decompilation/Manipulation

- Use Apktool to decompile into SMALI interpreter
  - Apktool d helloworld.apk
- Use text editor for small changes
  - E.g. Hello World becomes Hello Joe!
- Use Eclipse for larger changes
  - E.g. New functionality
- Reassemble with apktool
  - Apktool b helloworld.apk
  - Must sign before usable
- Digitally sign the new APK
  - Create signature
    - Jdk\jdk-ver\bin\keytool.exe -genkey -keystore keys/helloworld.keystore -alias HelloWorld -keyalg RSA -validity 10000
    - Create values when prompted
  - Sign APK
    - Jdk\jdk-ver\bin\jarsigner.exe -keystore keys\helloworld.keystore helloworld\dist\helloworld\HelloWorld
  - Signature not checked by Android; can deploy many ways

Mobile Device Ecosystem

- Lots of services used when mobile
  - Devices
  - Web servers
    - Mobile path may be less secure
    - FB mobile used to use http://m.facebook.com rather than https://...
  - Desktops
  - Mobile applications
  - Mobile sharing/syncing
    - DropBox
    - Google Drive
    - Etc.
- Analysis of Mobile device risks and threats are complex and require flexibility
- Devices accessing, processing and storing data outside your traditional boundaries
- Pentesting tricky as multiple factors involved, and you may not have or be able to get permission to pentest all of them

Mobile Device Pen Testing is complex

- Mobile Device Pen Testing building blocks:
  - Network pen testing
  - Wireless pen testing
  - Web pen testing
- Diversity of environment drives flexibility
- New tools emerging to help
- Old tools packaged to help
  - PWN Pad/Phone/PI

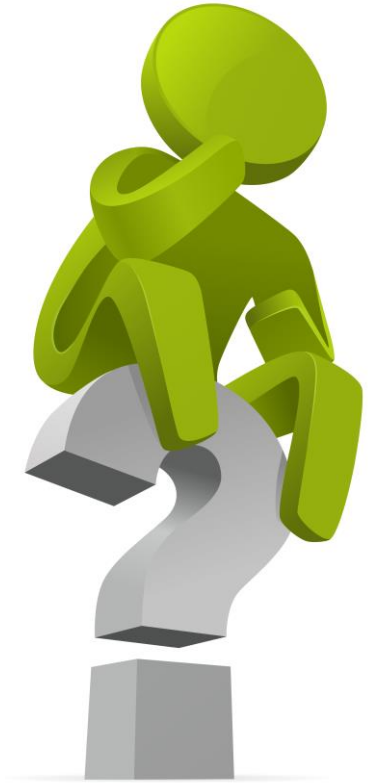
Pen Testing Mobile Applications

- Understand the application
  - Some are really a web site
  - Many are simply web pages rendered in an application
  - Others really are an application
- Questions to ask
  - Where is the data
  - How is it protected
  - How is access governed
  - Who owns what
  - Do you have a backup and can you restore it
- Once you know these, now go prove, or disprove
- What happens when you intercept and change parameters
  - Session timeout, cryptographic or other checks may thwart
- Can you replay transactions
  - Message Integrity Check may not prevent this
  - Transaction serialization helps
- Are apps susceptible to XSS
- Are apps susceptible to SQL Injection

Staying Current

- Field constantly changing
  - New devices
  - Constant OS Updates
  - New features = new vulnerabilities
- Get plugged into Security Data Feeds
  - US-CERT
  - Twitter:
    - ios security
    - android security
- Try your own hand pen testing known vulnerable apps
- CarnalOwnage Vulnerabile Android Apps
  - http://carnal0wnage.attackresearch.com/2013/08/want-to-break-some-android-apps.html

Questions?



Lee Neely
CISSP, CISA, CISM, CRISC, CCUV, OMBUDS
lkn@omp.net