Mobile Device Security
- Reading Material

Adam C. Champion and Dong Xuan
CSE 4471: Information Security

Based on materials from Tom Eston (SecureState),
Apple, Android Open Source Project, and William Enck (NCSU)
Organization

• Quick Overview of Mobile Devices
• iOS/Android Threats and Attacks
• iOS/Android Security
Overview of Mobile Devices

• Mobile *computers*:
  – Mainly smartphones, tablets
  – Sensors: GPS, camera, accelerometer, etc.
  – Computation: powerful CPUs ($\geq 1$ GHz, multi-core)
  – Communication: cellular/4G, Wi-Fi, near field communication (NFC), etc.

• Many connect to cellular networks: *billing system*

• Cisco: 7 billion mobile devices will have been sold by 2012 [1]
Organization

• Quick Overview of Mobile Devices
• iOS/Android Threats and Attacks
• iOS/Android Security
iOS/Android Malware

- iOS malware: very little
- Juniper Networks: Major increase in Android malware from 2010 to 2011 [18]
- Android malware growth keeps increasing ($$$)
- Main categories: [19]
  - Trojans
  - Monitoring apps/spyware
  - Adware
  - Botnets
- We’ll look at notable malware examples
iOS Malware

• Malware, “fake apps” have hit iOS too
  – iKee, first iPhone virus, “rickrolled” jailbroken iDevices [25]
  – Example “fake/similar” apps:
    • Temple Run: Temple Climb, Temple Rush, Cave Run
    • Angry Birds: Angry Zombie Birds, Shoot Angry Birds
    • Not to mention “walkthroughs,” “reference” apps, etc.
    • Google Play banned such apps…
  – iOS, Android hit with “Find and Call” app
    • SMS spammed contacts from central server
    • Removed from App Store, Google Play
Android: DroidDream Malware

- Infected 58 apps on Android Market, March 2011
- 260,000 downloads in 4 days
- How it worked:
  - Rooted phone via Android Debug Bridge (adb) vulnerability
  - Sent premium-rate SMS messages at night ($$$)
- Google removed apps 4 days after release, banned 3 developers from Market
- More malware found since
Android: Fake *Angry Birds Space*

- Bot, Trojan
- Masquerades as game
- Roots Android 2.3 devices using “Gingerbreak” exploit
- Device joins botnet

Source: [20]
Android: SMS Worm

• Students in previous information security classes wrote SMS worms, loggers on Android
  • Worm spreads to all contacts via social engineering, sideloading, etc.
  • Logger stored/forwarded all received SMS messages
    – Only needed SEND_SMS, RECEIVE_SMS, READ_SMS permissions
    – Can send 100 SMS messages/hour
    – One group put SMS logger on Google Play (removed it)
Android: Google Wallet Vulnerabilities (1)

- Google Wallet enables smartphone payments
  - Uses NFC technology
  - Many new mobile devices have NFC
- Some credit card info stored securely in secure element
  - Separate chip, SD card, SIM card
- Unfortunately, other data are not stored as securely
Android: Google Wallet
Vulnerabilities (2)

- Some information can be recovered from databases on phone: [21]
  - Name on credit card
  - Expiration date
  - Recent transactions
  - etc.

- Google Analytics tracking can reveal customer behavior from non-SSL HTTP GET requests

- NFC alone does not guarantee security
  - Radio eavesdropping, data modification possible [22]
  - Relay attacks, spoofing possible with libnfc [23]
Android: Sophisticated NFC Hack

- Charlie Miller’s Black Hat 2012 presentation: Nokia, Android phones can be hijacked via NFC [24]
  - NFC/Android Beam on by default on Android 2.3+, Android 4.0+
  - Place phone 3–4 cm away from NFC tag, other NFC-enabled phone
  - Attacker-controlled phone sends data to tag/device, can crash NFC daemon, Android OS
  - For Android 4.0–4.0.1, can remotely open device browser to attacker-controlled webpage
Organization

• Quick Overview of Mobile Devices
• iOS/Android Threats and Attacks
• iOS/Android Security
iOS System Architecture (1)

• Boot sequence:
  – Bootloader, kernel, extensions, baseband firmware all have cryptographic signatures
  – Root of trust: burnt into boot ROM at the factory
  – Each component’s signature is verified
  – If any signature doesn’t match, the “connect to iTunes” screen is shown
iOS System Architecture (2)

- Software updates
  - Cannot install older version of iOS on an iDevice; e.g., if device runs iOS 5.1.1, cannot install iOS 4
  - Device cryptographically “measures” components, sends to Apple install server with nonce, device ID
    - Nonce: value used only once
    - Prevents attacker from “replaying” the value
  - Server checks measurements; if allowed, server adds device ID to measurements, signs everything
iOS Apps and App Store

• All iOS apps signed by Apple (not developer)
• Third-party apps signed only after:
  – Developer ID verification (individual, company)
  – Review: bugs, work correctly (program analysis)
• Each app sandboxed in its own directory
  – Cannot communicate with other apps
  – Apps need signed “entitlements” to access user data
• Further app protection:
  – Address Space Layout Randomization (ASLR) for all apps
  – ARM eXecute Never (XN) bit set for all memory pages
iOS Data Protection Measures

• Each iDevice has hardware-accelerated crypto operations (AES-256)

• Effaceable Storage: securely removes crypto keys from flash memory
  – “Erase all content and settings” wipes user data using Effaceable Storage (locally or remotely)
  – Interact with mobile device management (MDM), Exchange ActiveSync servers
  – Developers can use APIs for secure file, database storage

• Passcodes
  – Admins can require numeric, alphanumerical, etc.
  – Wipe device after 10 failed login attempts
iPhone Configuration Utility
Miscellaneous iOS Security

• Built-in support for SSLv3, TLS, VPNs
• Extensive administrative controls:
  – Password policies
  – Disable device features, e.g., camera
  – Disable Siri
  – Remote wipe
• Apps can access contacts without permission (fixed in iOS 6)

Source: [8]
iOS Jailbreaking

• Circumvents Apple’s iOS security mechanisms
  – Violates iDevice’s terms of use
  – Allows installation of apps from alternative app stores, e.g., Cydia
  – Removes app sandbox
  – Usually replaces kernel with one accepting non-Apple signatures
  – Tools: redsn0w, Absinthe, etc.

• Legal in U.S. under DMCA 2010 exemption
Google Android Platform

• Android: Linux-based mobile handset platform
• Developed by Google, Open Handset Alliance for handset manufacturers
  – Includes T-Mobile, Sprint Nextel, Google, Intel, Samsung, etc. [29]
  – Free, open mobile handset platform for industry [30]
• Flagship: Google Nexus 4
Android Architecture
Android Features and Software

• Features
  – 3D: OpenGL ES 1.0
  – SQLite: Database engine
  – WebKit: Web browser
  – Dalvik: Register-based VM similar to Java VM [32]
  – FreeType: Bitmap and vector font rendering
  – Connectivity: Bluetooth, 802.11, GPS

• Core Applications
  – Email, SMS, calendar, Google apps, browser, etc.
  – Written in Java

• App Framework
  – Full access to same framework APIs
  – Architecture designed for component reuse

• Runtime
  – Core C++ library
  – Multiple Dalvik VMs run in a process, rely on Linux kernel for process isolation [32]
Android Security (1)

• Android built on Linux kernel, which provides
  – User permissions model
  – Process isolation
• Each app is assigned unique user/group IDs, run as a separate process \(\Rightarrow\) app sandbox
• System partition mounted read-only
• Android 3.0+ enables filesystem encryption using Linux dmcrypt (AES-128)
• Device admins can require passwords with specific criteria, remote wipe devices, etc.
Android Security (2)

• Android device administration (3.0+):
  – Remote wipe
  – Require strong password
  – Full device encryption
  – Disable camera
Other protection mechanisms:
- Android 1.5+: stack buffer, integer overflow protection; double free, chunk consolidation attack prevention
- Android 2.3+: format string protection, NX, null pointer dereference mitigation
- Android 4.0+: ASLR implemented
- Android 4.1+: ASLR strengthened, plug kernel leaks

Capability-based permissions mechanism:
- Many APIs are not invoked without permission, e.g., camera, GPS, wireless, etc.
- Every app must declare the permissions it needs
- Users need to allow these permissions when installing app
Android Security (4)

• All Android apps need to be signed: by the developer, not Google

• Google Play app store less regulated
  – Apps available rapidly after publishing
Android Device Diversity (1)

- Android runs on various devices
  - Different devices run different OS versions
  - Device manufacturers often add their own custom UIs, software
  - Mobile operators add their own software
  - Not all devices are updated to latest Android version!

- Security challenges…

Android devices accessing Google Play, August 2012. Some devices are not always updated to the latest version. These devices tend to have security vulnerabilities targeted by attackers.

Source: [12]
Android Device Diversity (2)

- Notice many Android devices are “orphaned” without major updates [13]
- Android developers need to secure their apps for many different devices…
The OpenSignalMaps Android app sees almost 4,000 types of device clients. Source: [14]
Rooting Android Devices

- Android device owners can often get root access to their devices
  - Process can be as simple as unlocking bootloader
  - Sometimes, exploit bugs to get root
  - Result: install OS of choice, bypass device/operator restrictions
  - Legal under 2010 DMCA exemption

- Security problems:
  - Voids device warranty (usually)
  - Circumvents app sandbox: root can modify any app’s files
  - Malware can root and own your device!

References (2)


References (3)

23. libnfc, http://www.libnfc.org


