Malware in iOS and Android
The Gathering Storm?
Introduction
About Me

• Security consultant
  • Android and iOS security testing
  • Researcher in Android malware
Agenda

• Introduction to Mobile Malware

• Introduction to Android and iOS

• Protection against Malware & Bypasses

• Consequences of Malware

• Mitigation
Malware in Mobile
Malware in Mobile

- “Traditional” malware vectors

1. Limit malware infection = Limit user install privileges
   Limit services exposed
Malware in Mobile

- “Traditional” malware vectors

2. Limit malware actions = Limit user privileges
Malware in Mobile

- “Traditional” malware vectors

3. Detection and removal = AV / Traffic analysis
Malware in Mobile

- Mobile malware vectors

1. Limit malware infection = Mobile vendor market place protections
   User review
Malware in Mobile

- Mobile malware vectors

2. Limit malware actions = Sandbox
Malware in Mobile

- Mobile malware vectors

3. Detection and removal = Vendor identification
Android & iOS
Android

- Unveiled 2007
- Open source license
- Applications can be downloaded from Google Play (formerly Android Market)
- Apps mostly written in Java
iOS

- Released by Apple in 2007
- Not licensed to non-Apple hardware
- Applications downloaded from the Apple app store
- Applications written in objC (and C)
Protection Against Malware
Protection from Malware - Android

- In the marketplace – Bouncer
  - Simulates running application

- On device - Sandboxing applications
  - Can still talk to other apps using their end-points
  - Restricts actions by permissions
  - Supports ASLR and DEP (but not implemented)

- Google can uninstall applications without user interaction
Protection from Malware - iOS

- Devices can only run code signed by Apple (unless Jail Broken)
- Applications are reviewed before being added to the app store
- On device security:
  - ASLR
  - W^X
  - Sandboxing
  - File encryption (when at rest)
- Apple can remove apps from devices
Bypassing Protection

- Charlie Miller’s iOS application
  - Post install code download
- MADDEN NFL 12 (IRC bot / SMS) for Android
  - Root exploit
  - Premium rate SMS
  - IRC bot communication
- Users trusting the developer
User permissive attacks

- Installing apps request permissions they want:
User permissive attacks

- Example by MWR – The “ricker” app

**PERMISSIONS**

This app can access the following on your phone:

- **Your messages**
  read SMS or MMS

- **Your personal information**
  read contact data

- **Services that cost you money**
  send SMS messages
User permissive attacks

• What the user sees:
User permissive attacks

- What is actually happening:
User permissive attacks

- Top 10 “hot” apps:
- Would you accept the following?
  - Internet Access?
  - Read your contacts?
  - Modify/Delete SD card contents?
  - Write / Modify your contacts?
Android - logs

- Android applications write to a common log file
- Any application with READ_LOGS permission can see all the logging from all applications.
- Includes Radio logs
- Android is found to be logging sensitive information:
  - Text messages
    ```
    05-10 13:45:48.447  1453  1542  I  RIL-MX : tokSmsRsp: dataLen = 81
    0791447728008000040C9144976815581200002150013154154006C2329BFDBE03+GC
    MT=66
    ```
  - Call information
    ```
    D/HTC_RIL (   70): RX::> +CLCC: 1,1,4,0,0,"07786172419",129\r\n0
    D/HTC_RIL (   70): (t=1336655587)<+CLCC: 1,1,4,0,0,"07786172459",129\r\n0
    ```
Zero permission app

- 0 permissions != No access

- By Default Android applications can access:
  - SD card (read only)
  - Public functions of other applications (i.e. Endpoints)
Zero permission app - example

- HTC Wildfire S – bought in October 2011
- Scanned with Mercury application
- Other application’s allowed communication to 130 database files that contained:
  - All my APN settings (including passwords!)
  - The current web page I'm browsing
  - My location (thanks weather app!)
  - My messages and attachments through HTC’s social application
  - All my download history
  - All my calendar history
Zero permission app - FOTA

• FOTA – Firmware Over the Air
• Checks site for updates of the phone’s firmware
• Installs if newer.
• Testing showed that:
  • These details were in a database
  • The database had SQL injection
  • The database was world writable
Consequences of Malware
Consequences

• How bad can malware really be?
  • User information can be sold
  • User credentials
  • Premium rate charges
  • Spam
  • DDos
  • What’s in your phone?
Mitigation
Mitigation

- Is there a solution?
- One opinion:

"Who here has an iPhone? Who here has a BlackBerry? Who here uses Gmail? Well, you're all screwed"

- Julian Assange
Mitigation

- It’s the responsibility of the User

- Control what is on the device
  - Sensitive information
  - Connections
  - Applications
Mitigation

- It’s the responsibility of the Developer
- Need to develop for the platform
  - Data in memory
  - Secure storage
  - Secure connections
  - Application interaction control
Mitigation

- It’s the responsibility of the Company
- Unique to mobile approach
- iOS has basic management controls through enterprise solutions.
- Android has Device Policy
Mitigation

• Device review
  • Tailored to mobile problems

• Creation of “trusted devices” that uses a “trusted market”
  • Review of devices and the applications and services installed
  • Source code review of applications
Questions?