DENIM GROUP

build | integrate | secure

Putting the Smart in Smartphones: Security Testing Mobile Applications

My Background

- Dan Cornell, founder and CTO of Denim Group
- Software developer by background (Java, .NET, etc)
- OWASP San Antonio, Global Membership Committee

- Denim Group
  - *Build software with special security, performance, reliability requirements*
  - *Help organizations deal with the risk associated with their software*
    - Code reviews and application assessments
    - SDLC consulting
    - Secure development training – instructor-led and eLearning

Agenda

- Introduction and Overview
- Mobile Application Threat Model
- Testing Approaches
- Example Application
- Data at Rest
- Data in Motion
- Tainted Inputs
- Conclusions / Questions

Smart Phones, Dumb Apps

- Lots of media focus on device and platform security
  - *Important because successful attacks give tremendous attacker leverage*
- Most organizations:
  - *Accept realities of device and platform security*
  - *Concerned about the security of their custom applications*
  - *Concerned about sensitive data on the device because of their apps*
  - *Concerned about network-available resources that support their apps*
- Who has mobile application deployed for customers?
- Who has had mobile applications deployed without their knowledge?
  - **$!%$# marketing department…*

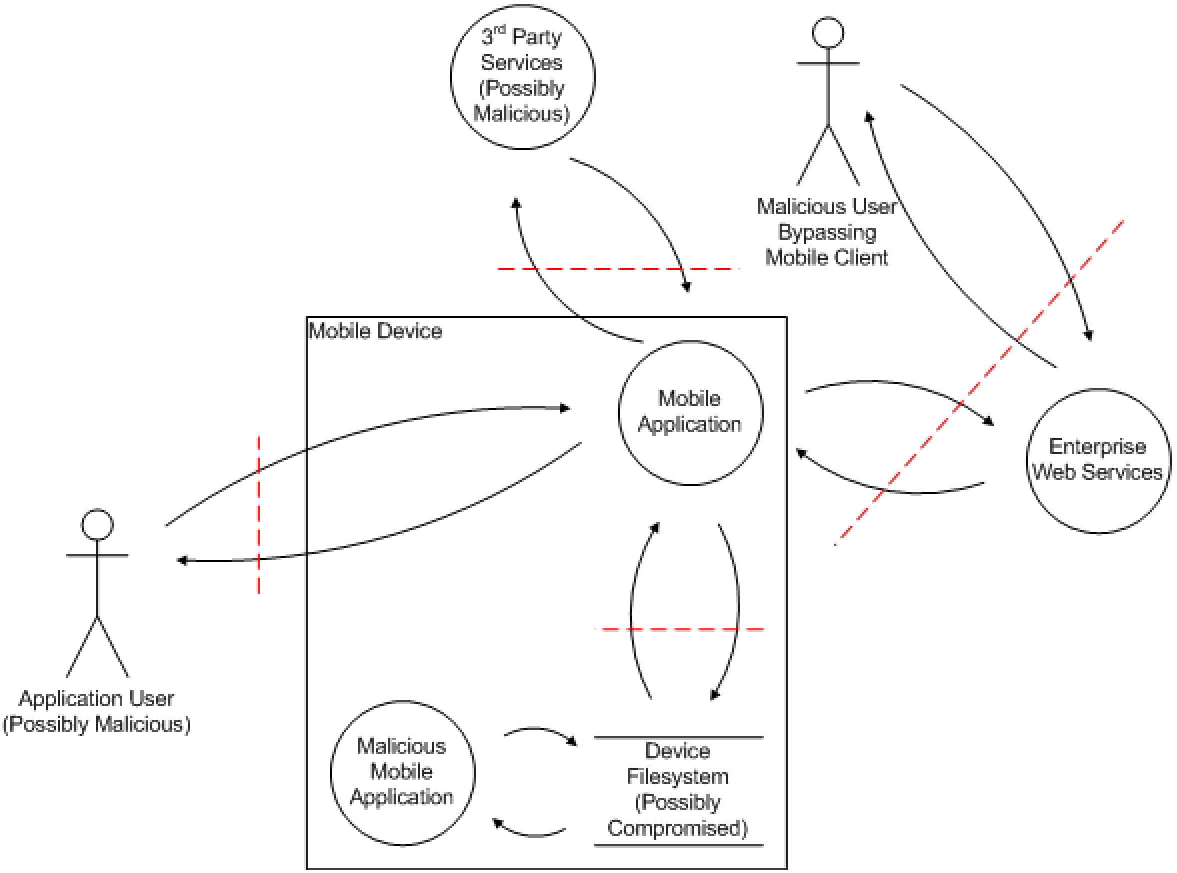
Some Assumptions for Developers

- Smartphone applications are essentially thick-client applications
  - *That people carry in their pockets*
  - *And drop in toilets*
  - *And put on eBay when the new iPhone comes out*
  - *And leave on airplanes*
  - *And so on…*

- Attackers will be able to access:
  - *Target user (victim) devices*
  - *Your application binaries*

- What else should you assume they know or will find out?

Generic Mobile Application Threat Model


3rd Party Services (Possibly Malicious)
Malicious User Bypassing Mobile Client
Mobile Device
Mobile Application
Enterprise Web Services
Application User (Possibly Malicious)
Malicious Mobile Application
Device Filesystem (Possibly Compromised)

Testing the Security of Mobile Applications

- IMPORTANT: It is really the system as a whole you care about
  - *Application plus…*
  - *3rd party web services*
  - *Enterprise services*
  - *And so on*

- The most "interesting" weaknesses and vulnerabilities we find are in mobile applications' interactions with supporting services

- Mobile applications are different than web applications
  - *Can't just fire up an automated scanner and turn up a bunch of SQL injection and XSS vulnerabilities*
  - *Usually…*

Testing the Security of Mobile Applications

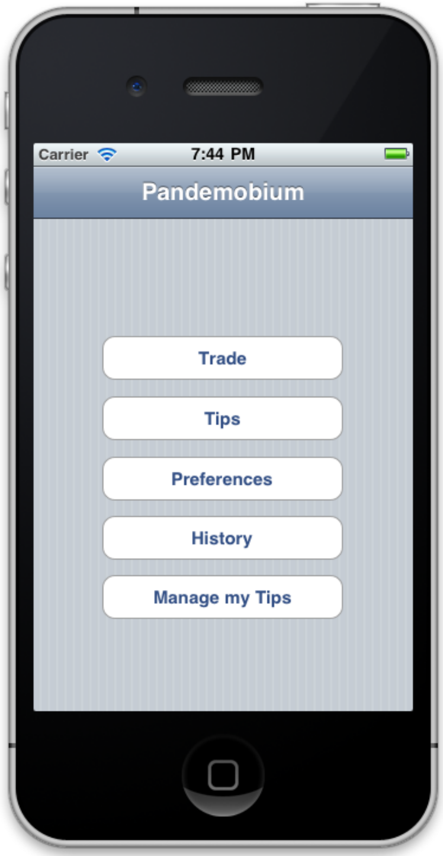
Type of Analysis	Activities
Static Analysis	
Source Code	Source code scanning Manual source code review
Binary	Reverse engineering
Dynamic Analysis	Debugger execution Traffic capture via proxy
Forensic Analysis	File permission analysis File content analysis

Testing the Security of Mobile Applications

- Know you enemy
  - *So you can properly characterize risk*
- How can attackers gain unauthorized access?
  - *Attacker steals or accesses a lost device*
  - *Malicious application*
  - *Attacker reverse engineers an application to access corporate resources*
  - *And so on…*

Pandemobium Stock Trader Application

- Android and iOS versions
- Functionality
  - *Log in*
  - *Track stock tips*
  - *Make stock trades*
  - *Get stock tips*
  - *Share stock tips*



Let’s Take Apart Some Apps: Android

- Example of static binary analysis
- Application structure
  - *AndroidManifest.xml*
  - *assets/*
  - *res/*
  - *classes.dex*

- axml2xml.pl
  - http://code.google.com/p/android-random/downloads/detail?name=axml2xml.pl
- dedexer
  - http://dedexer.sourceforge.net/
- dex2jar
  - http://code.google.com/p/dex2jar/
- JD-GUI
  - http://java.decompiler.free.fr/
- SQLite Browser
  - http://java.decompiler.free.fr/

Let's Take Apart Some Apps: iOS

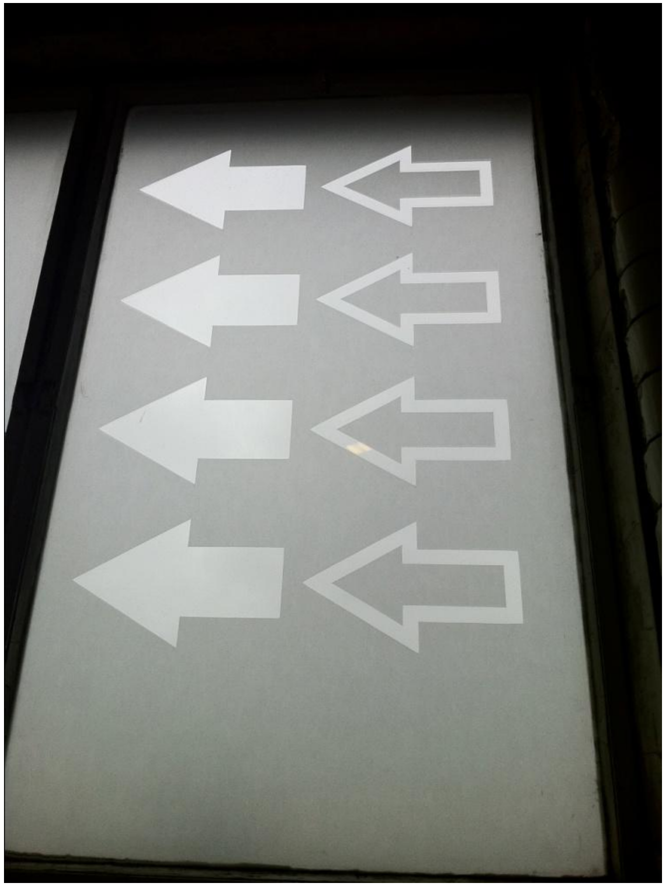
- More static binary analysis
- Application structure
  - *Application binary*
  - *plist files*
  - *Other resources*

- otool
  - http://developer.apple.com/library/mac/#documentation/Darwin/Reference/ManPages/man1/otool.1.html
- plutil
  - http://developer.apple.com/library/mac/#documentation/Darwin/Reference/ManPages/man1/plutil.1.html
- IDA-PRO
  - http://www.hex-rays.com/idapro/
- iPad File Explorer
  - http://www.ipadfileexplorer.com/

Identifying Potential Storage Issues

- Static analysis
  - *Identify functions that store data locally on the device*
- Forensic analysis
  - *Run the application and look at artifacts it creates*

Data in Motion



- 3^{rd} Party Services
- Enterprise Services

Identifying Services In Use

- Look for URL connections
- Look for network connections
- Look for web controls

Tainted Inputs

- Mobile Browser Content Handling

Android: Identifying Content Handlers

- Look in AndroidManifest.xml
- Look for <intent-filter> tags:

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="the_scheme" />
</intent-filter>
```

- But what apps export intents?
  - http://www.openintents.org/

iOS: Identifying Content Handlers

- Look in Info.plist
- Look for <key>CFBundleURLSchemes</key>

```
<array>
    <dict>
        <key>CFBundleURLSchemes</key>
        <array>
            <string>the_scheme</string>
        </array>
    </dict>
</array>
```

- But what apps handle custom schemes?
  - *http://handleopenurl.com/*

Testing the Security of Content Handlers

- How to reach them?
  - *Get a user to click: <a href=“the_scheme://stuff?param=value” />*
  - *Get a user to visit a malicious web page:*

    <iframe src=“the_scheme://stuff?param=value” />
- Approaches:
  - *Fuzzing*
  - *Targeted attacks*

But How Bad is SQL Injection in Mobile Apps?

- Probably not as bad as SQL injection for web applications
  - *Probably*
- Remember DREAD:
  - *Damage Potential*
  - *Reproducibility*
  - *Exploitability*
  - *Affected Users*
  - *Discoverability*

The End

Conclusions and Questions

Dan Cornell

dan@denimgroup.com

Twitter: @danielcornell

www.denimgroup.com

www.smartphonesdumbapps.com

(210) 572-4400