Understanding the Threat Profile of Mobile Apps

Session 11

Wednesday, November 20, 2013

11:15 AM - 12:30 PM

Ming Chow

Tufts University

The Conference On
Mobile & Smart Device Security

MIS®
TRAINING INSTITUTE

Key Points

1. Common vulnerabilities and flaws in mobile applications
2. The security features/defenses major platforms provide
3. What attackers of mobile app flaws actually exploit
4. The impact of mobile app attacks
5. App security recommendations/solutions
6. To confirm most of what Charlie Miller said in his keynote yesterday in this room, November 19, 2013

Preliminaries: Why Mobile Is Different

- High value
- Device can be easily lost or broken
- Has good computational power
- It is “always on” (including networking), even when you are not interacting directly with device
- Constraints: battery, screen size, input
- Features: GPS, accelerometer, compass, NFC
- New security model including app distribution
- No authorized method for gaining administrative access by default
- Hence, some old security models (e.g., anti-virus) do not work

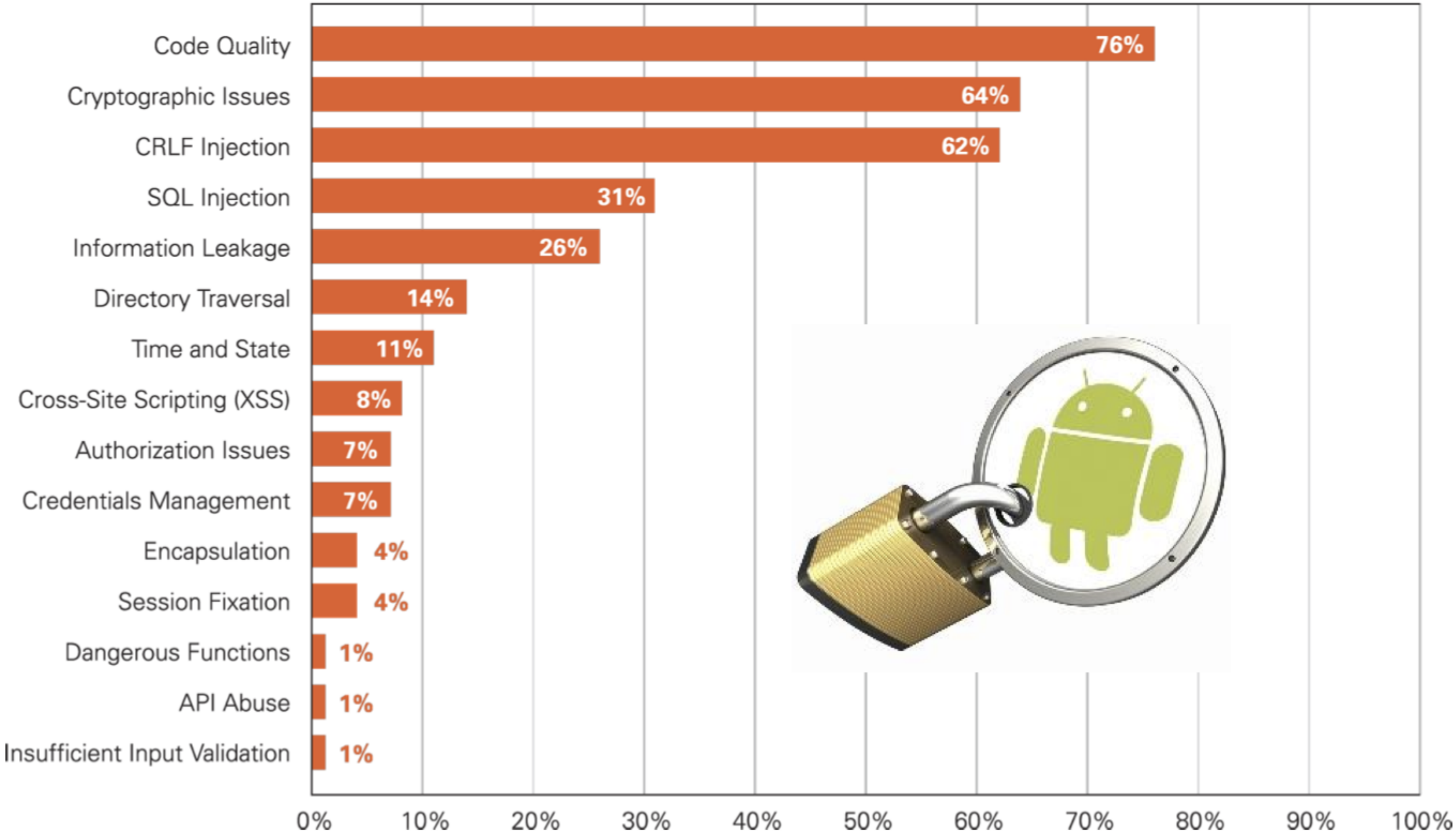
What Will Not Be Discussed

- Mobile malware
- Mobile web apps and the mobile browser (that was my talk yesterday)
- Jailbreaking
- Exploit development
- Forensics

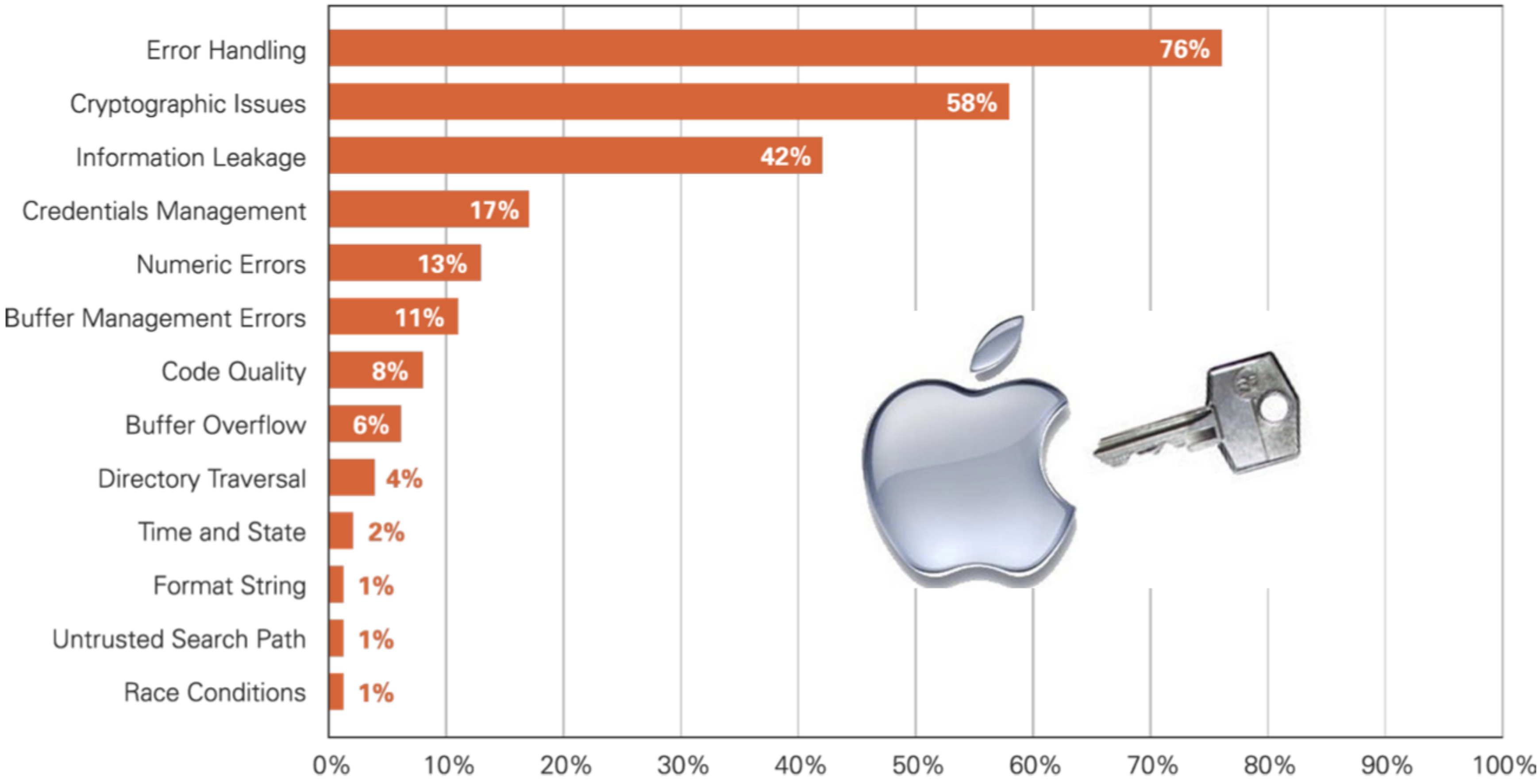
It Isn’t Pretty for Mobile Apps

- See Veracode’s **State of Software Security Report: Volume 5**…
- …or **Chris Wysopal’s** presentation **We See the Future and It’s Not Pretty: Predicting the future using vulnerability data (SOURCE Boston Conference 2013)**
- Compelling numbers
  - One of the key findings: “Cryptographic issues affect a sizeable portion of Android (64%) and iOS (58%) applications.”

Android Vulnerability Prevalence (Percentage of Applications Affected)



iOS (ObjectiveC) Vulnerability Prevalence (Percentage of Applications Affected)



Vulnerability Distribution for Mobile Platforms (Share of Total Vulnerabilities Found)

Android		iOS		Java ME	
CRLF Injection	37%	Information Leakage	62%	Cryptographic Issues	47%
Cryptographic Issues	33%	Error Handling	20%	Information Leakage	47%
Information Leakage	10%	Cryptographic Issues	7%	Directory Traversal	3%
SQL Injection	9%	Directory Traversal	6%	Insufficient Input Validation	2%
Time and State	4%	Buffer Management Errors	3%	Credentials Management	<1%

UPDATED from Veracode (November 15, 2013; thanks to Chris Wysopal): Android Vulnerability Prevalence

Code Quality	94%
Cryptographic Issues	78%
CRLF Injection	76%
Information Leakage	39%
SQL Injection	37%
Time and State	31%
Directory Traversal	29%
Credentials Management	20%
Insufficient Input Validation	17%
Authorization Issues	17%
Potential Backdoor	8%
Cross-Site Scripting (XSS)	3%
Dangerous Functions	1%

UPDATED from Veracode (November 15, 2013; thanks to Chris Wysopal): iOS Vulnerability Prevalence

Error Handling	92%
Cryptographic Issues	85%
Credentials Management	51%
Information Leakage	48%
Code Quality	14%
Buffer Management Errors	11%
API Abuse	11%
Numeric Errors	10%
Directory Traversal	9%
Buffer Overflow	3%
CRLF Injection	1%
SQL Injection	1%

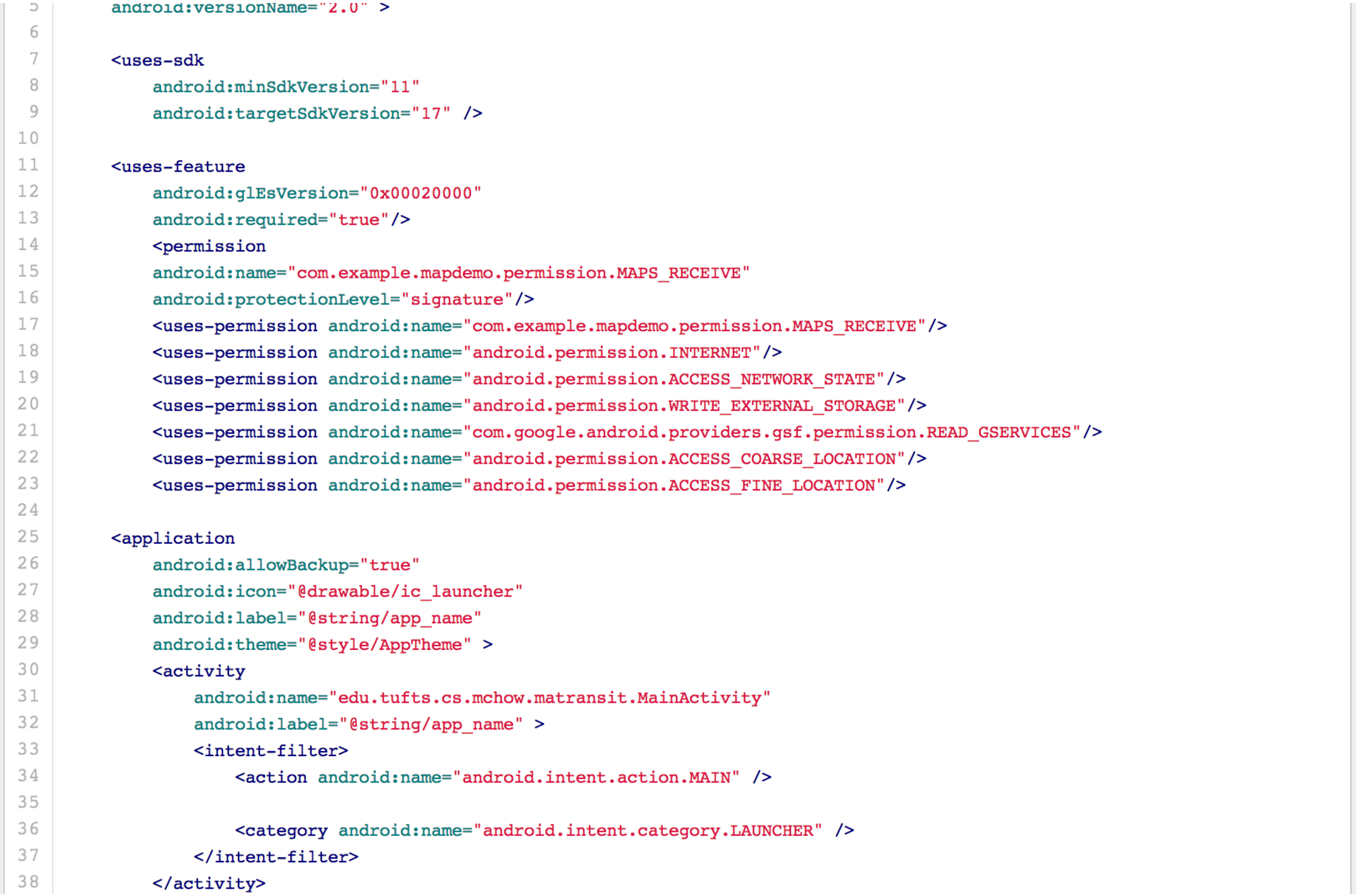
Android Security Model

- Documentation: http://source.android.com/devices/tech/security/
- The application sandbox (by default)
  - "System assigns a unique user ID (UID) to each Android application and runs it as that user in a separate process."
  - "Applications cannot interact with each other and applications have limited access to the operating system."

Android Security Model (continued)

- Least privilege-based permissions to protected APIs, system resources (e.g., camera, GPS, Internet)
  - Apps can only access a limited range of system resources *by default*
  - Permissions must be stated in `AndroidManifest.xml` file for app
- Permissions check on sensitive data and inputs including calendar and personal information (user grants access)
- No Security Manager (legacy code), no Java Sandbox
- Can make system calls (via NDK)
- Apps can access the kernel, drivers, syscalls, etc. --if device is rooted
- Address Space Layout Randomization (ASLR) supported in Android 4.0

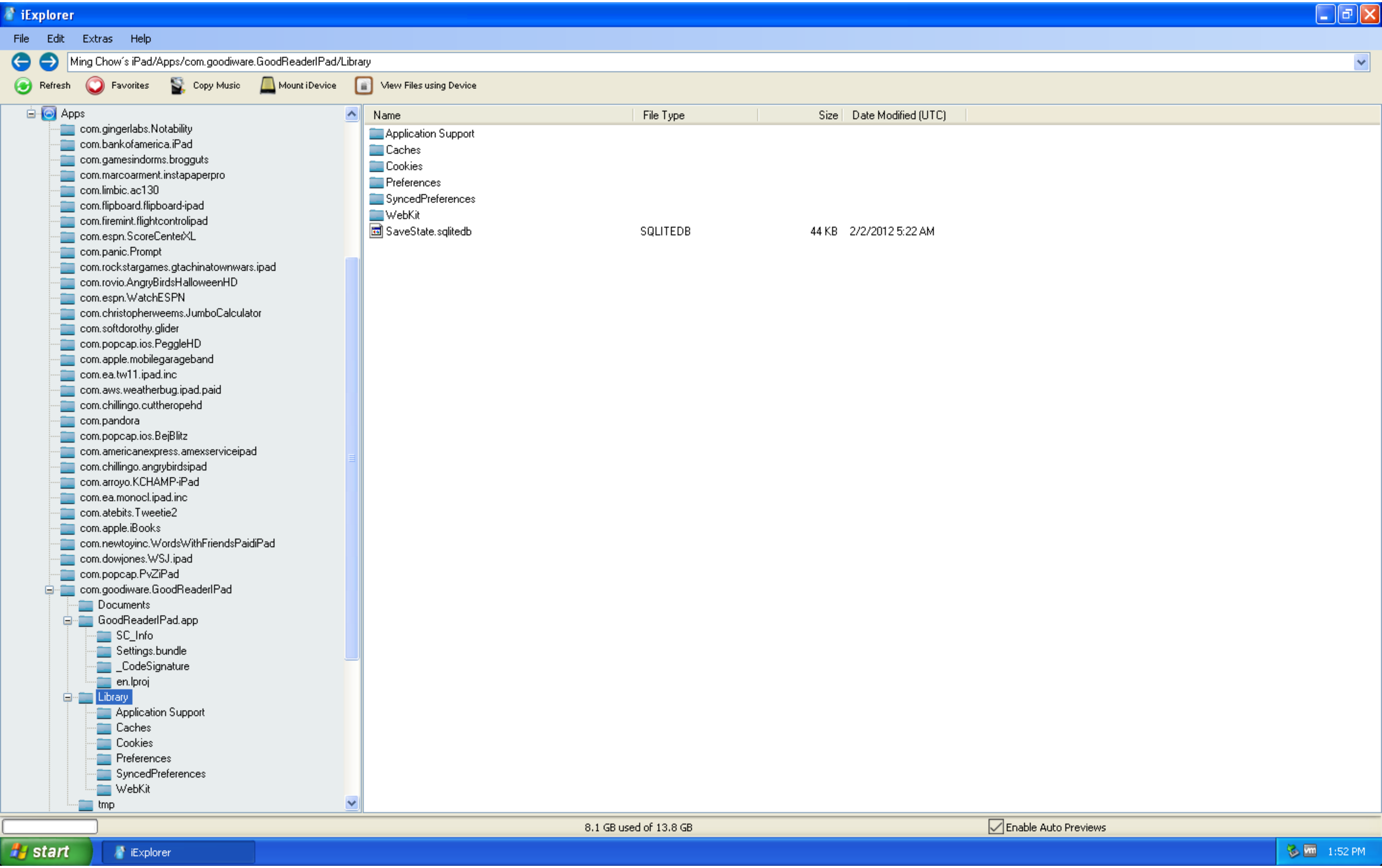
Example of an AndroidManifest.xml file (my app)



```
    android:versionName="2.0" >

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="17" />

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true"/>
        <permission
        android:name="com.example.mapdemo.permission.MAPS_RECEIVE"
        android:protectionLevel="signature"/>
        <uses-permission android:name="com.example.mapdemo.permission.MAPS_RECEIVE"/>
        <uses-permission android:name="android.permission.INTERNET"/>
        <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
        <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
        <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
        <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
        <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="edu.tufts.cs.mchow.matransit.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

iOS Security Model

- Documentation: http://www.apple.com/ipad/business/docs/iOS_Security_Oct12.pdf
- Address Space Layer Randomization (ASLR) since iOS 4.3
- Apps run as the same user, but…
  - Apps must be signed by Apple (code signing)
    - App ID, development certificate (public/private keys), production certificate (public/private keys) all required
  - Apps are given a unique ID and directory by Apple
- Sandboxed apps: apps cannot access data from other apps
- To use Apple services including Game Center, iCloud, Passbook, Push Notification Service, development and production certificates (public/private keys) also required



Ming Chow's iPad/Apps/com.goodiware.GoodReaderIPad/Library

```
Apps
  com.gingerlabs.Notability
  com.bankofamerica.iPad
  com.gamesindorms.brogguts
  com.marcoarment.instapaperpro
  com.limbic.ac130
  com.flipboard.flipboard-ipad
  com.firemint.flightcontrolipad
  com.espn.ScoreCenterXL
  com.panic.Prompt
  com.rockstargames.gtachinatownwars.ipad
  com.rovio.AngryBirdsHalloweenHD
  com.espn.WatchESPN
  com.christopherweems.JumboCalculator
  com.softdorothy.glider
  com.popcap.ios.PeggleHD
  com.apple.mobilegarageband
  com.ea.tw11.ipad.inc
  com.aws.weatherbug.ipad.paid
  com.chillingo.cuttheropehd
  com.pandora
  com.popcap.ios.BejBlitz
  com.americanexpress.amexserviceipad
  com.chillingo.angrybirdsipad
  com.arroyo.KCHAMP-iPad
  com.ea.monocl.ipad.inc
  com.atebits.Tweetie2
  com.apple.iBooks
  com.newtoyinc.WordsWithFriendsPaidiPad
  com.dowjones.WSJ.ipad
  com.popcap.PvZiPad
  com.goodiware.GoodReaderIPad
    Documents
    GoodReaderIPad.app
      SC_Info
      Settings.bundle
      _CodeSignature
      en.lproj
    Library
      Application Support
      Caches
      Cookies
      Preferences
      SyncedPreferences
      WebKit
    tmp
```

Name	File Type	Size	Date Modified (UTC)
Application Support			
Caches			
Cookies			
Preferences			
SyncedPreferences			
WebKit			
SaveState.sqlitedb	SQLITEDB	44 KB	2/2/2012 5:22 AM

8.1 GB used of 13.8 GB

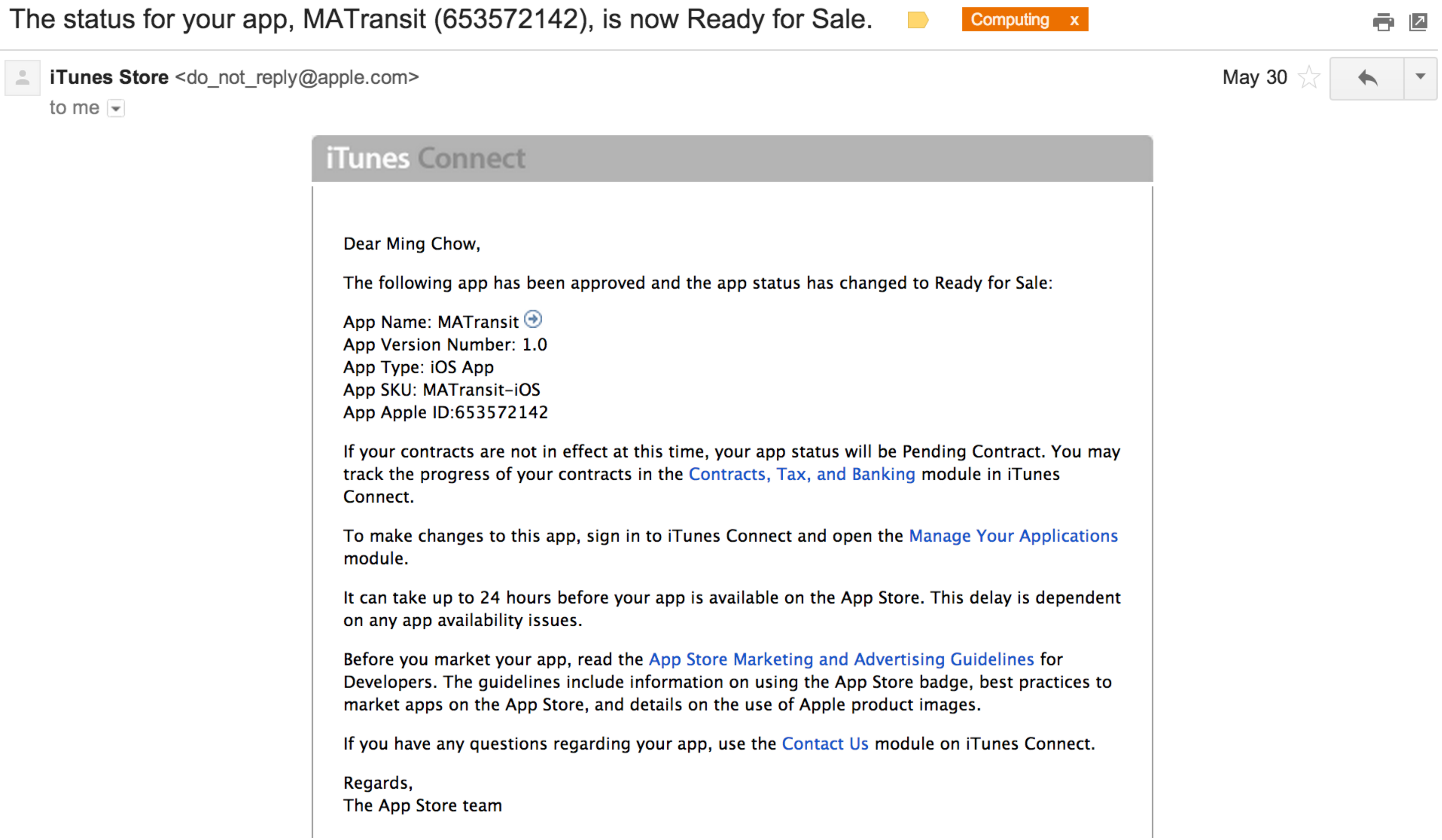
☑ Enable Auto Previews

iOS Developer Program Registration

- $99.00 (not including taxes) per year
- "Provide basic personal information, including your legal name and address. If you're enrolling as a company/organization, we'll need a few more things, like your legal entity name and D-U-N-S® Number, as part of our verification process."
- You can revoke and change development and production app certificates
- Devices must be whitelisted for testing
- The moral of the story: "Apple knows who you are"

iTunes App Store App Submission Process

- Must pass manual content review by Apple
- Static analysis used in app review process
- Developer(s) will receive either approved or reject email
- There is App Review Board to appeal rejections
- More: https://developer.apple.com/appstore/guidelines.html
- You can be banned



The status for your app, MATransit (653572142), is now Ready for Sale.

iTunes Store <do_not_reply@apple.com>

to me

May 30

Dear Ming Chow,

The following app has been approved and the app status has changed to Ready for Sale:

App Name: MATransit
App Version Number: 1.0
App Type: iOS App
App SKU: MATransit-iOS
App Apple ID:653572142

If your contracts are not in effect at this time, your app status will be Pending Contract. You may track the progress of your contracts in the Contracts, Tax, and Banking module in iTunes Connect.

To make changes to this app, sign in to iTunes Connect and open the Manage Your Applications module.

It can take up to 24 hours before your app is available on the App Store. This delay is dependent on any app availability issues.

Before you market your app, read the App Store Marketing and Advertising Guidelines for Developers. The guidelines include information on using the App Store badge, best practices to market apps on the App Store, and details on the use of Apple product images.

If you have any questions regarding your app, use the Contact Us module on iTunes Connect.

Regards,
The App Store team

Google Play App Store Submission Process

- Registration: a one time $25 registration fee
- No identity verification necessary.
- Dynamic analysis is used in app review
- No app review board
- No manual content review
- No device whitelisting for testing; just take USB cable, transfer APK to device, and run APK to install app onto phone assuming developer mode is enabled
- Lost the private key for signing Android APK? You're out of luck, cannot release updates to app.

Malicious App Campaigns

- From **Mobile Exploit Intelligence Project by Guido and Arpaia (SOURCE Boston Conference 2012)**
- iTunes App Store: 0
- Google Play: 30
- The moral of the story: “Say what you will about police states, but they have very little crime.”
- Could be worse: third-party app stores
  - For jailbroken iOS devices
  - For Android (e.g., Amazon, many in Asia)
  - Many more malicious app campaigns

Crypto and Transport Layer Issues

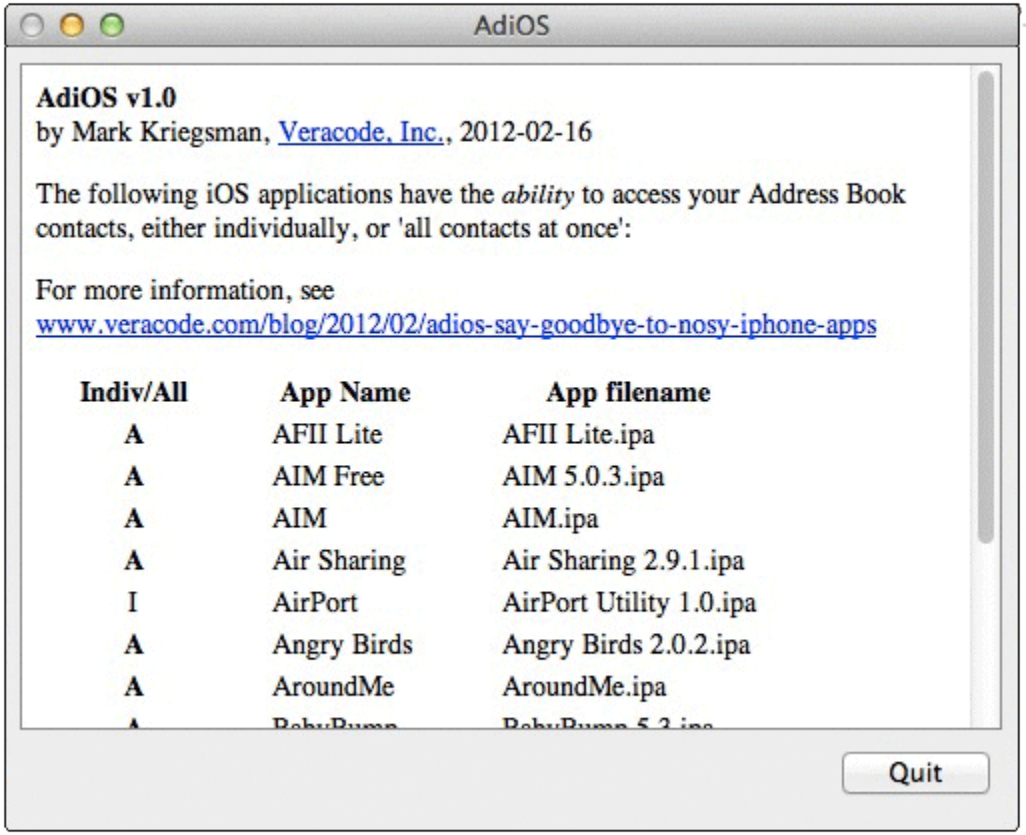
- Hard-coding cryptographic keys directly into mobile app (e. g., to a *.plist file in iOS, to AndroidManifest.xml)
- Not using HTTPS (e.g., for GET and POST API calls)
- Not hashing passwords; stored in a plaintext or transported in the clear
- Recent examples:
  - Tube Map Live Underground (BlackBerry and Android App)
  - Super Backup (Android App)
  - https://www.appthority.com/risky-apps/risky-apps-private-user-data-exposed-by-super-backup-and-tube-map-live-underground
- Rolling your own crypto, snake oil

More Information Leakage From Apps

- Your entire address book was sent to Path (iOS App)
  - http://mclov.in/2012/02/08/path-uploads-your-entire-address-book-to-their-servers.html
  - Path was fined $800K by the FCC for collecting information without users' (especially children's) consent
  - Now an iOS app must ask for permission to use your contacts
- Path was not the only app that took your address book
  - http://allthingsd.com/20120215/following-path-address-book-uproar-many-apps-clean-up-their-acts/

Veracode’s AdiOS

http://www.veracode.com/blog/2012/02/adios-say-goodbye-to-nosy-iphone-apps/



AdiOS v1.0
by Mark Kriegsman, Veracode, Inc., 2012-02-16

The following iOS applications have the *ability* to access your Address Book contacts, either individually, or 'all contacts at once':

For more information, see
www.veracode.com/blog/2012/02/adios-say-goodbye-to-nosy-iphone-apps

Indiv/All	App Name	App filename
A	AFII Lite	AFII Lite.ipa
A	AIM Free	AIM 5.0.3.ipa
A	AIM	AIM.ipa
A	Air Sharing	Air Sharing 2.9.1.ipa
I	AirPort	AirPort Utility 1.0.ipa
A	Angry Birds	Angry Birds 2.0.2.ipa
A	AroundMe	AroundMe.ipa
A	BabyBump	BabyBump 5.3.ipa

What Are Pentesters Doing and Finding

- Wi-Fi attacks
- Phishing (email and SMS) and social engineering
- Running remote attacks
- Running mitmproxy
- Using Android (default or jailbroken) devices
- Decompiling Android APKs
- Fuzz testing
- Findings: information leakage, privilege escalation

Advanced Research

- Executing special Unstructured Supplementary Service Data (USSD) codes to remote wipe Android devices (Borgaonkar, 2012)
- Twitter SMS Spoofing via phone number, assuming no lock (Rudenberg, 2012)
- Android NFC vulnerabilities (Miller, 2012; Wall of Sheep 2013)
- Facebook, LinkedIn, and Dropbox app authentication keys stored in plain text on iOS (2012)
- Zeus-in-the-mobile (Zitmo) Android malware (2012)
- Juice-jacking (Wall of Sheep 2011; Lau, Jang, and Song, 2013)
- Android master key vulnerability (Forristal, 2013)
- Rooting SIM cards (Nojl, 2013)
- Google Chrome and iOS Safari vulnerabilities (Mobile Pwn2Own 2013)

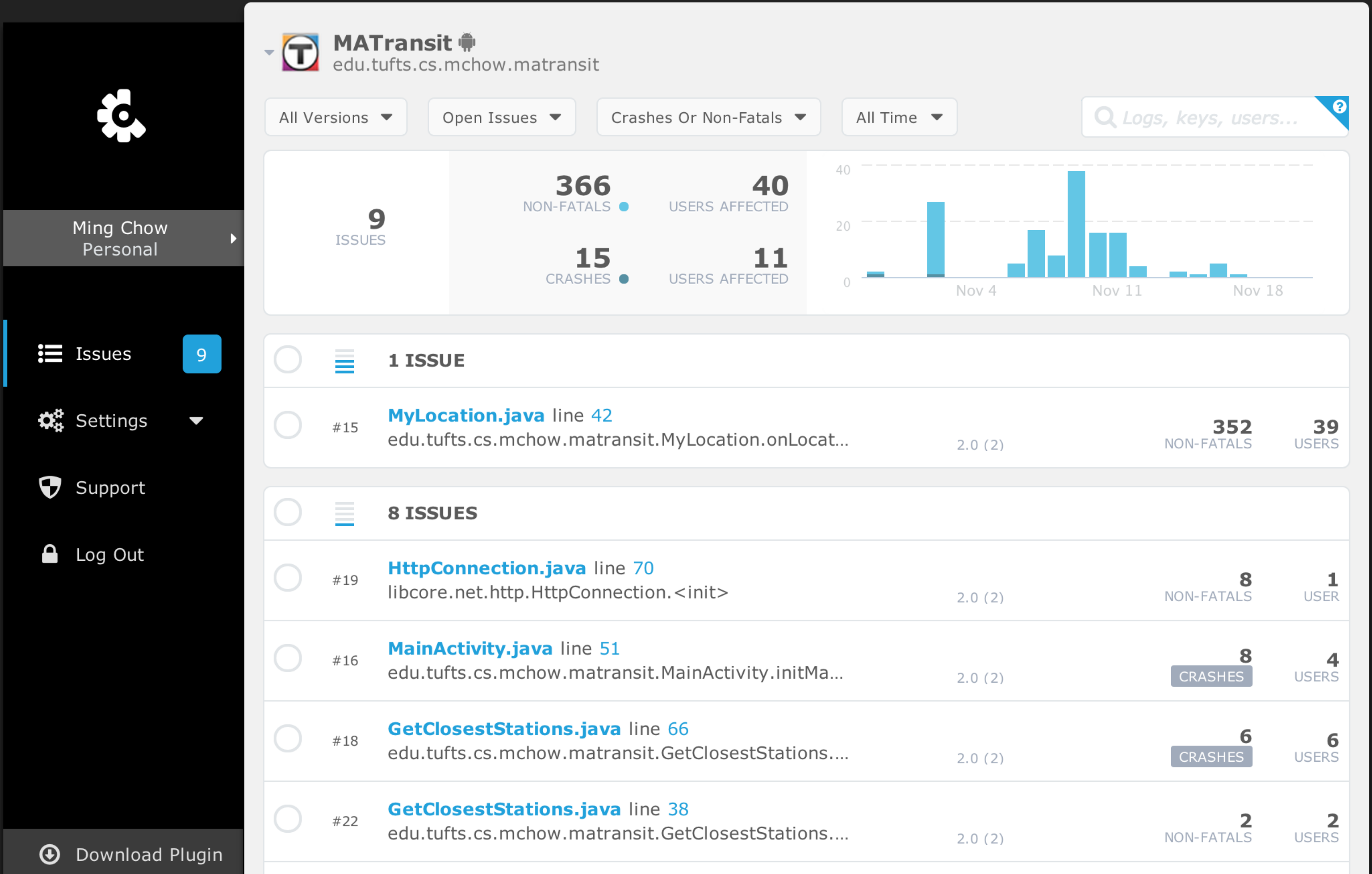
Impact, The Loss (as of now)

- Credentials (Facebook, banking)
- Photos
- Address books, contacts
- Your location
- Installation of malware to steal SMS and call logs, etc.

Recommendations and Tools

- Tools
  - mitmproxy (http://mitmproxy.org/)
  - Crashlytics (for developing Android and iOS apps; https://crashlytics.com/)
  - Binary static analysis tools, dynamic analysis (e.g., Veracode)
- Readings:
  - https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/
  - https://www.isecpartners.com/media/11991/isec_securing_android_apps.pdf
  - http://www.veracode.com/blog/2013/08/developers-guide-to-building-secure-mobile-applications-infographic/
  - http://www.praetorian.com/blog/develop-secure-mobile-apps-studying-vulnerable-android-ios-mobile-web-apps

Crashlytics



MATransit
edu.tufts.cs.mchow.matransit

9 ISSUES

366 NON-FATALS ● 40 USERS AFFECTED

15 CRASHES ● 11 USERS AFFECTED

1 ISSUE

#15 **MyLocation.java** line 42
edu.tufts.cs.mchow.matransit.MyLocation.onLocat... 2.0 (2) 352 NON-FATALS 39 USERS

8 ISSUES

#19 **HttpConnection.java** line 70
libcore.net.http.HttpConnection.<init> 2.0 (2) 8 NON-FATALS 1 USER

#16 **MainActivity.java** line 51
edu.tufts.cs.mchow.matransit.MainActivity.initMa... 2.0 (2) 8 CRASHES 4 USERS

#18 **GetClosestStations.java** line 66
edu.tufts.cs.mchow.matransit.GetClosestStations.... 2.0 (2) 6 CRASHES 6 USERS

#22 **GetClosestStations.java** line 38
edu.tufts.cs.mchow.matransit.GetClosestStations.... 2.0 (2) 2 NON-FATALS 2 USERS

The Third-Party Server

- Even if your app passes static and dynamic analysis, and have no crashes, there may be serious problems on the server end
- Servers will always be targeted
- Example: my app messagehub for Android and iOS (not on app stores) and the corresponding messagehub web application has GET and POST JSON APIs
- January 2013: ESPN ScoreCenter app (iOS) cross-site scripting (XSS) and cleartext password vulnerabilities http://research.zscaler.com/2013/01/mobile-app-wall-of-shame-espn.html

The Future

- Mobile is here to stay
- The good news: vendors (e.g., Apple, Google, Samsung) respond to concerns
- The bad news:
  - Fragmentation of devices; some devices may not be able to update apps or the OS (think carriers)
  - No end in sight to the mobile malware problem on Android; 718,000 malicious apps and counting
  - Online banking moving to mobile
  - Sophistication necessary to find mobile exploits. Look at what pentesters are finding vs what is actually getting exploited
- Problems and cost of impact bound to be greater later

References

- http://www.csoonline.com/article/742371/the-department-of-homeland-security-and-its-obsolete-android-os-problem?source=rss_malware_cybercrime
- http://www.veracode.com/blog/2013/10/our-apps-are-our-digital-lives/
- http://www.veracode.com/blog/2012/07/how-sally-got-owned-an-illustrated-example-of-how-piracy-can-endanger-your-mobile-device/
- http://www.veracode.com/blog/2011/04/mobile-app-privacy-continued/
- http://securitywatch.pcmag.com/mobile-apps/317788-mobile-threat-monday-london-transit-app-and-android-backup-app-leak-personal-info
- http://www.veracode.com/blog/2012/02/adios-say-goodbye-to-nosy-iphone-apps/
- http://tuftsdev.github.io/DefenseOfTheDarkArts/lecture_notes/predicting_the_future_veracode.pdf
- http://www.trailofbits.com/resources/mobile_eip-04-19-2012.pdf
- http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-repeating-history.pdf
- http://www.bluecoat.com/sites/default/files/documents/files/BC_2013_Mobile_Malware_Report-v1d.pdf
- http://gcn.com/blogs/pulse/2013/06/gartner-mobile-big-data-advanced-targeted-attacks-shape-threat-landscape.aspx
- http://www.symantec.com/threatreport/topic.jsp?aid=analysis_of_mobile_threats&id=threat_activity_trends
- https://www.veracode.com/images/pdf/soss/state-of-software-security-report-volume5.pdf
- http://www.darkreading.com/vulnerability/top-mobile-vulnerabilities-and-exploits/240144260
- http://www.securityweek.com/hackers-demo-two-iphone-exploits-safari-mobile-pwn2own
- http://blog.trendmicro.com/trendlabs-security-intelligence/trend-micro-solution-for-vulnerability-affecting-nearly-all-android-devices/
- http://www.blackhat.com/us-13/briefings.html#Lau
- http://blog.trendmicro.com/trendlabs-security-intelligence/exploiting-vulnerabilities-the-other-side-of-mobile-threats/
- https://viaforensics.com/mobile-security/chained-vulnerabilities-firefox-android-pimp-browser.html
- https://viaforensics.com/mobile-security/mobile-security-challenges-wall-street-journal.html
- http://www.wallofsheep.com/pages/nfc-security-awareness-project
- http://bluebox.com/corporate-blog/bluebox-uncovers-android-master-key/
- http://about-threats.trendmicro.com/us/security-roundup/2013/2Q/mobile-threats-in-full-throttle/