SMARTPHONE APPS ARE NOT THAT SMART
ME?

VULNEX:  www.vulnex.com
Blog:     www.simonroses.com
Twitter:  @simonroses
TALK OBJECTIVES

- Apps are the new Web
- Peek into current state of Apps security on Markets
- Bugs will be revealed but not the victims
DISCLAIMER

All Apps are considered safe until proven guilty by a security review
AGENDA

1. IT’S ALL ABOUT APPS
2. APPS RISKS
3. CASE STUDIES
4. SECURITY DEVELOPMENT TIPS
5. CONCLUSIONS
1. IT’S ALL ABOUT APPS
1. WHY SMARTPHONE APPS?

• IDC predict smartphone sales will rise to 982 million in 2015

• Morgan Stanley Research estimates sales of smartphones will exceed those of PCs in 2012

• Average installed apps is 65, but average consumer uses only 15 apps per week

• In 2011 an average of 701 apps were launched in the UK version of Apple App Store every day!!!

• An App for almost anything...
1. APPS BY THE NUMBERS

DOWNLOADED APPS

- IPhone (February 2011) 18 Billions
- Android (December 2011) 10 Billions
- WP7: ¿?
# 1. SMARTPHONE DEVELOPMENT

<table>
<thead>
<tr>
<th>IPhone</th>
<th>Android</th>
<th>WP7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed</td>
<td>Java</td>
<td>.NET</td>
</tr>
<tr>
<td>Native</td>
<td>Objective-C</td>
<td>C&lt;sup&gt;(1)&lt;/sup&gt;</td>
</tr>
<tr>
<td>Web</td>
<td>Action Script, HTML, CSS, JavaScript</td>
<td>Action Script, HTML, CSS, JavaScript</td>
</tr>
<tr>
<td>Scripting</td>
<td>Ruby</td>
<td>Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl, and shell</td>
</tr>
<tr>
<td>3º Party / Free / Commercial</td>
<td>Java, c#</td>
<td>Visual Basic, c#</td>
</tr>
</tbody>
</table>

---

<sup>(1)</sup> Parts of C in Java Apps / Full C apps at platform level  
<sup>(2)</sup> Currently only Microsoft but is coming
2. APPS RISKS
2. APPS SECURITY RESEARCH

• Rules of Engagement

➢ 100 apps analyzed from official markets
➢ Each app one hour review top or less
➢ Different categories analyzed:

• Security
• Social networking
• Communications
• Servers
• Finance
• Media
• Productivity
• Travel
2. APPS ANALYZED BY NUMBERS

- Social Networking ➔ +2 million
- Finance ➔ 500000
- Productivity ➔ 10 million
- Security ➔ 5 million
- Media ➔ 100000
- Travel ➔ 5 million
2. OWASP MOBILE PROJECT

- OWASP started in 2010 a mobile security project

- Goal: To give developers and security pros resources to secure mobile Apps

- Milestones:
  - OWASP Top 10 Mobile Risks
  - Security development & testing guides
  - OWASP GoatDroid Project
2. OWASP MOBILE TOP 10 RISKS

- Insecure Data Storage
- Weak Server Side Controls
- Insufficient Transport Layer Protection
- Client Side Injection
- Poor Authorization and Authentication
- Improper Session Handling
- Security Decisions Via Untrusted Inputs
- Side Channel Data Leakage
- Broken Cryptography
- Sensitive Information Disclosure
2. VULNEX APPS RISKS

- Weak Crypto
- Excessive Permissions
- Mixing Social Features
- Clear Text Secrets
- Data Validation
- Insecure Channels
- Privacy Issues
- Debug Code Enabled
- Phone Back home
- Cross Site Scripting
- Dynamic SQL
3. CASE STUDIES
3. CLEAR TEXT SECRETS

- App fails to protect sensitive information, credentials
- OWASP Mobile: M1- Insecure Data Storage

<table>
<thead>
<tr>
<th>Data Storages</th>
<th>IPhone</th>
<th>Android</th>
<th>WP7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Data</td>
<td></td>
<td>Shared Preferences</td>
<td>Isolated Storage</td>
</tr>
<tr>
<td>SQLite Databases</td>
<td></td>
<td>Internal Storage</td>
<td>Network Connection</td>
</tr>
<tr>
<td>Logs</td>
<td></td>
<td>External Storage</td>
<td></td>
</tr>
<tr>
<td>Network Connection</td>
<td></td>
<td>SQLite Databases</td>
<td></td>
</tr>
<tr>
<td>Property List (plist)</td>
<td></td>
<td>Network Connection</td>
<td></td>
</tr>
<tr>
<td>XML</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3. CLEAR TEXT SECRETS EXAMPLE:
CREDENTIALS MANAGER (CVE-2011-1840)

```xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<int name="CountDownValue" value="45" />
<int name="PasswordLength" value="10" />
<boolean name="EncryptDb" value="true" />
<boolean name="AutoDestruction" value="false" />
<boolean name="ListView" value="true" />
<boolean name="ReadablePassword" value="true" />
<int name="Order" value="0" />
<boolean name="OnlyNumberPassword" value="false" />
<boolean name="CountDown" value="true" />
<boolean name="SecurePassword" value="false" />
<boolean name="Inactivity" value="true" />
<boolean name="startupPassword" value="true" />
<string name="Password">2222</string>
<int name="InactivityValue" value="150" />
</map>
```
3. CLEAR TEXT SECRETS MITIGATION

• Use encryption and platform secure features (Information at rest)

• Set correct file permissions

• Avoid to save data to external / public storage areas (mostly SD Cards)
3. INSECURE CHANNELS

- App sends data over network without encryption (HTTP vs. HTTPS)
  - Watch out for credentials
  - PII data (chats, Facebook, etc.)

- When using encrypted channels, perform certification validation

- OWASP Mobile: M3- Insufficient Transport Layer Protection
3. INSECURE CHANNELS EXAMPLE:
SOCIAL NETWORKING

POST http://example.com/api HTTP/1.1
Host: example.com
User-Agent: 
Content-Length: 298
Accept: */*
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-us
Connection: keep-alive
Proxy-Connection: keep-alive

{"version":"0.7.1","requests": ["getSession","passcode": "3c2bc9cdc59d5ee5437a6a58df3a35", "seed": "ABsXShfLj1TQ2izMeM+4Kt66og4t18e7meD4SU6cPgwDg="","email": "test","timestamp": 1330638622,"application_key": "MDI3MDfZmZjU4MGExNWM0YmEyYjjA5M2RkODlmMjg0MTU6MC43NzQ4ODAwMCAxMjc1NDcyNjg2"], ["getVersion", []]}

VULNEX
3. INSECURE CHANNELS MITIGATION

- Encrypt sensitive data going out device (Protect information in transit)

- Applies to any type of connection
3. DEBUG ENABLED

- App ships to market with logging or debugging features enabled

- Helps attacker to learn Apps internal

- OWASP Mobile: M8- Side Channel Data Leakage
import android.util.Log;

public final class DebugLog
{
    private static boolean mLoggingEnabled = 1;

    public static int d(String paramString1, String paramString2)
    {
        int i = 0;
        if (mLoggingEnabled)
        {
            String str = paramString2;
            i = Log.d(paramString1, str);
        }
        return i;
    }
}
3. DEBUG ENABLED EXAMPLE: SERVER

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.package" android:versionName="1.0" android:versionCode="1">
  <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="android.permission.ACCESS_WIFI_STATE"/>
  <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
  <uses-permission android:name="android.permission.WAKE_LOCK"/>
  <application android:debuggable="true" android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name="com.example.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>
    <activity android:name="com.example.AdActivity"/>
    <service android:name="com.example.MyService"/>
  </application>
  <uses-sdk android:minSdkVersion="4"/>
</manifest>
```
3. DEBUG ENABLED EXAMPLE: FINANCE

<table>
<thead>
<tr>
<th>L..</th>
<th>Time</th>
<th>PID</th>
<th>Application</th>
<th>Tag</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>jdwp</td>
<td></td>
<td>received file descriptor 10 from ADB</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>ddm-heap</td>
<td></td>
<td>Got feature list request</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... dalvikvm</td>
<td></td>
<td>GC freed 2765 objects / 220768 bytes in 84ms</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... TermsActivity</td>
<td></td>
<td>onCreate</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... loadTermsInfo</td>
<td></td>
<td>loadTermsInfo</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... loadTermsInfo</td>
<td></td>
<td>StoredFileName-&gt;/storage/emulated/0/WWW_...</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... loadTermsInfo</td>
<td></td>
<td>StoredTimeStap-&gt;634364532369648955</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... TermsActivity</td>
<td></td>
<td>onResume</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>com.b... HttpClient</td>
<td></td>
<td>get url: http://...</td>
</tr>
<tr>
<td>W</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>ExpatReader</td>
<td></td>
<td>DTD handlers aren't supported.</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td>(requestOp...</td>
<td></td>
<td>RequestOperation successful.</td>
</tr>
<tr>
<td>D</td>
<td>03-01 23:28:3... 452</td>
<td></td>
<td></td>
<td></td>
<td>local TermsInformationDownloaded.</td>
</tr>
</tbody>
</table>
3. DEBUG ENABLED MITIGATION

• For debug code:
  – What data is saved to logs?
  – Where is the data saved to?

• Android: Eclipse turns off debuggable by default on release
3. DATA VALIDATION

- App fails to perform appropriate data validation
- Accounts for many common risks
- OWASP Mobile: M4- Client Side Injection
public void deleteCaseValue(String paramString) {
    SQLiteDatabase localSQLiteDatabase = this.mDb;
    String str = "DELETE FROM case_values WHERE _id = " + paramString;
    localSQLiteDatabase.execSQL(str);
}
3. CROSS SITE SCRIPTING (XSS) EXAMPLE:
3. CROSS SITE SCRIPTING (XSS) EXAMPLE, IN CASE YOU MISSED IT
3. DATA VALIDATION EXAMPLE: MEDIA
3. DATA VALIDATION MITIGATION

- Validate data for:
  - Valid
  - Safe
  - Length

- For SQL queries use prepared statements

- Validate (sanitize) and escape data before render for web Apps

- Use white list approach instead black list approach. Check out OWASP ESAPI libraries
3. PII COMPROMISE

- App can collect plenty of PII information
  - User: username, contacts, bookmarks
  - Device: S.O. ver, device name, IMEI, IMSI, kernel version, UUID
  - General info: geolocalization

- OWASP Mobile Risk Classification: M8 – Side Channel Data Leakage
3. PII COMPROMISE MITIGATION

• Apps don't need to collect all they can, just what they need

• If collecting PII:
  – Where is that info going?
    • Log files
    • Data storages
    • Network

  – Protect it:
    • Transit
    • At Rest
3. 3RD PARTY LIBRARIES INTEGRATION

- App integrates 3rd party libraries:
  - Facebook
  - Greendroid
  - Android.ads
  - Apache
  - google.android.apps.analytics
  - Json
  - Mozilla
  - Javax
  - xmlrpc.android
  - slf4j
3. 3RD PARTY LIBRARIES INTEGRATION MITIGATION

• If using 3rd party libraries, use proven libraries

• What info are these libraries collecting?

• Do we really need social networking libs integrated into our finance apps?
3. PERMISSIONS

• It’s important to understand App permissions

• App can compromise device security and user pocket
3. PERMISSIONS EXAMPLE - SEXYPIC
3. PERMISSIONS EXAMPLE - DROIDDREAM
3. PERMISSIONS MITIGATION

• User: Apply common sense

• Developer: Don’t abuse on permissions request (overprivileged)
3. WEAK CRYPTO

- Incorrect use of crypto libraries
- Implementing custom bad ass crypto algorithm
- M9 - Broken Cryptography
3. WEAK CRYPTO EXAMPLE - SECURITY

Default password in code

Encrypt pwd with MD5 (no salt)

Pwd stored in text file with world perms

File stored in SD card

http://www.md5-hash.com/
3. WEAK CRYPTO MITIGATION

• Use proven crypto libraries and read documentation!

• Forget about your own crypto

• If using SHA1 or MD5 for passwords apply salt, even better use SHA-256

• If using SHA1PRNG set the seed
3. HARDCODED CREDENTIALS

- App contains credentials embedded in code
- Easy to spot by attackers
- OWASP Mobile: M10- Sensitive Information Disclosure
3. HARDCODED CREDENTIALS EXAMPLE: SERVER

```java
private static final String LOGIN_URL = "http://example.com/login";
private static final String PASSWORD = "tetra1404ad3772";
public static final String PREFS_NAME = "application_prefs";
```
3. HARD CODED CREDENTIALS MITIGATION

• Easy, don't write credentials into code files 😊

• What happens when the credentials change? You need to upload a new version on the app!

• Credentials need to use secure data storages
4. SECURE DEVELOPMENT TIPS
4. TIPS (I)

- Apps need to pass Software Security Assurance practices
- Threat Modeling your Apps
- Understand platform and Apps risks
- Professional security reviews are expensive but small ISV and single developers can use available resources
4. TIPS (II)

• You can add jailbreak detection but is a losing race.
  ➢ Android:
    • Check if /system/app/Superuser.apk exist ?
    • Check if com.noshufou.android.su package exist ?
    • Can we write directly to /data/data ?

  ➢ IPhone
    • Call fork()
    • Check if /Applications/Cydia.app exist ?

  ➢ WP7
    • Allowed by Microsoft, [http://labs.chevronwp7.com/](http://labs.chevronwp7.com/)

• Code Obfuscation
4. SECURITY RESOURCES

- Iphone

- Android

- WP7

- OWASP Mobile Security Project
5. CONCLUSIONS
5. SUMMARY

• Apps are a non stop business

• Apps are really interesting for attackers, millions of potential targets
  – Malware authors

• With a short sample of analyzed Apps some interesting bugs were discovered

• Different classes of vulnerabilities but more exist than showed here
5. NEXT STEPS

• Automatize Apps analysis
  – Static Analysis
  – Dynamic Analysis

• Study cross platform technologies and their impact on security
  – Managed Apps (Mono)
  – Are bug cross platform?
5. Q&A

• Please fill out the black hat feedback form

• Thanks!