Security and Privacy issues in iOS and Android Apps

Praveen Nallasamy
OWASP Leader, NYC Chapter
Praveen.nallasamy@owasp.org
Praveen.nallasamy@gmail.com
www.praveennallasamy.com
OWASP NY/NJ Local Chapter Meeting

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.
Agenda

- About This Talk
- Need for Mobile Security
- Top 5 Issues
  1. Device Tracking
  2. Insecure Storage
  3. Insecure Communication
  4. Excessive Permissions
  5. Web Based vulnerabilities
- Questions
  - Love to answer all of them at the end of the presentation
About This Talk

- Focuses on iOS and Android apps
- iOS refers to iPhone / iPad / any other iDevice from Apple
- Have done over 35 pen tests on mobile apps in both these platforms over the last 10 months
- Found some interesting security and privacy issues
- Some results are shared here
- Top 5 high risk issues frequently seen
Need for Mobile Security

- Explosion in mobile devices – smartphones, tablets, many more form factors
  - Rapid increase in mobile malware on both iOS and Android platforms
  - Bad guys have learned how to monetize every piece of information about us
Need for Mobile Security

- Our Mobile devices know more about us than anyone else
  - What we browse /like /do/watch/listen/search for
  - Where we go
  - Personally Identifiable Information
  - Mobile banking, e-commerce
  - Social Networking

- For corporations it’s about branding and law
  - Security flaws in banking and payment apps
  - Law suits on Apple, Pandora, Weather channel for privacy invasion
  - Congress has introduced a bill on how Geo Location services can be used in Mobile devices
  - For Marketers and Advertisers mobile devices and their apps have information that is worth a gold mine
  - A lot of attention from Media
Top 5 Security and Privacy Issues

Top 5 Issues seen during our mobile penetration tests:

1. Device Tracking
2. Insecure Storage
3. Insecure Communication
4. Excessive Permissions
5. Web Based vulnerabilities
1. Device Tracking

What are Device Identifiers?
Think of them as similar to the VIN number of a vehicle.

1. UDID (Unique Device Identifier) – Apple Serial Number
2. IMEI (International Mobile Equipment Identity) Number – Unique GSM number, applicable to Android and iOS as long as it is on a GSM phone

- Most Apps collect at least one of the device identifiers
- App owners collect them
- Third party ad-networks that display banner ads inside the apps collect them
- Device IDs are collected because they now uniquely identify every device and the behavior of its user
1. Device Tracking

Collecting Device Identifiers – Privacy Risk

- Devices IDs can be deemed as personal information
  
  - Grey Area - Debate as to whether or not this it is legal.
  
1. Device Tracking

Collecting Device Identifiers – Privacy Risk

- A historical perspective
  - Lawsuit on Intel a decade back for sharing serial numbers of processors with developers
  - Apple has brought it back
    - [http://digitalarchive.gsu.edu/cgi/viewcontent.cgi?article=1850&context=gsulr&sei-redir=1#search="intel chipping away boundaries](http://digitalarchive.gsu.edu/cgi/viewcontent.cgi?article=1850&context=gsulr&sei-redir=1#search="intel%20chipping%20away%20boundaries)
1. Device Tracking

Collecting Device Identifiers – Privacy Risk

- User Tracking
  - Device IDs can be considered as personal information
  - Dangerous in combination with GPS and apps with Social Media permissions
  - Social Media plugins enable users share their personal information with Apps
  - Third Party ad network libraries collect data from multiple apps using UDID to get a better behavioral profile about its users...for more targeted ads
  - Track targeted users with GPS (if app has GPS permissions)
    - Eg. Bob is at Times Square, likes rock music (learnt from FB), pop an ad for Hard Rock café

- Besides user tracking
  - Can learn the device specifications based on the IMEI number
1. Device Tracking

Example: Online tools are available that provide device specifications using IMEI number.
1. Device Tracking

Example: Android app transmitting IMEI number in clear text.
1. Device Tracking

Example: iOS app transmitting UDID in clear text.
1. Device Tracking

Solution:
- Don’t use device identifiers
- When there is a business case use Salted Hashes of the Device IDs (MD5, SHA)
2. Insecure data storage

Storing sensitive information (i.e. PII, Passwords etc) local to the phone or device.

Sensitive Data Could Include
- Username / Passwords
- Device IDs
- PII, SSN, Health Information
- Application Configuration
- Credit card numbers

Why not?
- Phones can be lost or stolen
- Trivial to recover data if device is:
  - “jailbroken”
  - Rooted or
  - Not password protected
- In other cases partial or full recovery of data may be still possible if there is physical access to the device
2. Insecure data storage

Types of files where sensitive data may be present on Android apps

- Database files – SQL Lite files, *.db files
  - SQL Lite Browser or Command line SQL Lite can be used to view them
- Regular ASCII files, log files and Binary Files
  - Text Editors and Hex Editors can be used to view them
2. Insecure data storage

Location of sensitive files on Android Apps
- `cd /data/data/<app name>` – all application specific data files are located here:
2. Insecure data storage

Example: Android app storing device ID in plain text in a XML file.
2. Insecure data storage

Location of sensitive files on iOS devices

- **Application Specific Cache**
  - `~/Library/Application Support/iPhone Simulator/4.2/Applications/<app id>`

- **Snapshot Cache**
  - `~/Library/Application Support/iPhone Simulator/4.2/Applications/<app id>/Library/Caches/Snapshots/`

- **Temp Files Cache (PDF, xls, doc, jpeg etc.)**
  - `~/Library/Application Support/iPhone Simulator/4.2/Applications/<app id>/Documents/`

- **Clipboard Cache**
  - `~/Library/Application Support/iPhone Simulator/4.2/Library/Caches/com.apple.UIKit.pboard`

- **Key Stroke Cache**
  - `~/Library/Application Support/iPhone Simulator/4.2/Library/Keyboard/dynamic-text.dat`
2. Insecure data storage

Types of files where sensitive data maybe present on iOS Apps

- Database files – SQL Lite files, *.db files
- BinaryCookies
- Property List (.plist files)
2. Insecure data storage

Example: iOS app storing Admin passwords in clear text.
2. Insecure data storage

Example: iOS app storing Active Directory passwords.
2. Insecure data storage

Example: iOS – Binary Cookies.
2. Insecure data storage

Example: iOS – Binary Cookies can be viewed using hex editors
2. Insecure data storage

Example: iOS – Property List Files can be viewed using any editor.
2. Insecure data storage

Solution

- Avoid local storage inside the device for sensitive information
- If local storage is “required” encrypt data securely and then store
- Use the Crypto APIs provided by Apple and Google
- Avoid writing custom crypto code – prone to vulnerability
3. Insecure Communication

Risks:
- Mobile Internet is an insecure channel
- Public Wifi hotspots are open unsecured networks
  - Hotspots at Coffee Shops, Book Stores, Airports
  - Plenty of open source tools available to sniff from open wireless networks
- Firesheep addon for Firefox makes it easier
  - Grabs your Social Media and other web passwords with one click
- Face Sniffer app for Android is the Firesheep version for Mobile devices to sniff passwords from open wireless networks
3. Insecure Communication

- It is possible to throw a fake GSM signal
- Chris Paget demonstrated a fake GSM tower during DefCon 2010 that costed about $1500
- It is called IMSI catcher
- An attacker can throw up a fake ATT / T-Mobile signal a few feet away.
- Your phone would connect to his tower since it would have a stronger signal than the nearest cell phone tower.
- All data that is sent unencrypted can be read by the attacker.

3. Insecure Communication

A portable IMSI catcher that is usually available for law enforcement.
3. Insecure Communication

Solution:
- For practical purposes, let's accept all our mobile communication channels may be insecure
- Use SSL
- SSL / TLS is required for:
  - Login Credentials
  - PII
  - Credit Card numbers, SSN
  - Device Identifiers (UDID, IMEI etc.)
  - Any potentially sensitive information
3. Insecure Communication

Example: An Android App sending Twitter credentials in clear text.
3. Insecure Communication

Example: Android App sending IMEI number in clear text.
3. Insecure Communication

Example: iOS App sending user credentials using Basic Auth without SSL.
3. Secure Communication

Example: iOS App with UDID Transmission in clear text.
4. Excessive Permissions

Risks:

- Excessive permissions can turn users away
- Can steal customer data
- Can invade users’ privacy
- Can incur costs to the users
  - Eg. Wallpaper application having access to GPS
  - Eg. Notepad application with permission to send SMS, make calls

Applications should only have the required permissions to work
4. Excessive Permissions

What permissions Apps look for?
- Access to GPS
- Camera
- Contacts
- Access to make calls, send SMS
- System Settings
4. Excessive Permissions

Testing:

- **Android**
  - Transparent - all the permissions an app has access to
  - Permissions can found during Installation
  - AndroidManifest.XML file contains the permission details
  - Can be viewed anytime under Managing Applications

- **iOS**
  - Not so transparent
  - If the app has access to GPS there is prompt after installation. Other permissions are unknown to the user.
  - Assessor can look at source code for different frameworks used (eg. CoreLocation for GPS)
  - Walk through the source code to identify what permission an app has access to
4. Excessive Permissions

Example: Android – Wallpaper App having access to phone contacts
4. Excessive Permissions


```xml
</intent-filter>
</receiver>
</application>
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-feature android:name="android.hardware.camera" />
</manifest>
```
4. Excessive Permissions

Example iOS – Location Permission after installation of the app.
4. Excessive Permissions

Example: iOS App transmitting UDID along with GPS location.
4. Excessive Permissions

Examples: iOS App Sharing GPS location with other users and discloses what time the user was at certain location.
4. Excessive Permissions

Examples: iOS App constantly recording GPS location of the user
4. Excessive Permissions

Example: iOS – GPS and other permissions can be identified through the source code.
5. Web Application vulnerabilities

Why are they applicable?

- Apps talk to the server using HTTP
- Each app is like a browser
- Understands HTML, Javascript and other web application technologies
- Most web application vulnerabilities are applicable in the mobile application context
5. Web Application vulnerabilities

- SQL Injection
- Verbose Errors
- XSS
- Insecure Direct Object References
- Forceful Browsing
- Weak Authentication and Session management
- Security mis-configurations
Closing

- A few years ago people used to say Google knows more about us than anyone else. Today our mobile devices know more about us than Google.
- Mobile web application assessment must be integrated into SDLC programs and assessed on a periodic basis.
- Mobile web application should be evaluated from both a security and privacy perspective.
Questions
Questions / Comments / Feedback

praveen.nallasamy@owasp.org
praveen.nallasamy@gmail.com