Simón Roses Femerling

- Founder & CEO, VULNEX: [www.vulnex.com](http://www.vulnex.com)
- Blog: [www.simonroses.com](http://www.simonroses.com)
- Twitter: @simonroses
- Former Microsoft, PwC, @Stake
- DARPA Cyber Fast Track (CFT) grant to research on software security
- Speaker: Black Hat, RSA, OWASP, SOURCE, DeepSec, AppSec, Microsoft TECHNET
TALK OBJECTIVES

• Apps are the new Web (and doing same old mistakes)

• Secure Software Development Practices are a must

• Attacking Mobiles & Apps is much easier than you think!
AGENDA

1. Apps Development
2. Write Secure Apps
3. Conclusions
1. Apps Development
1. WHY SMARTPHONE APPS?

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

- 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

Opportunity Abounds
Number of mobile apps on different platforms

Apple
No. of Apps
700,000+

Android (Google)
No. of Apps
700,000+

Microsoft Windows
No. of Apps
125,000

Blackberry 10
No. of Apps
70,000

DOWNLOADED APPS

- IPhone (2013)
  50+ Billions

- Android (May, 2013)
  48+ Billions
1. PHONES CONTAIN YOUR LIFE

- Emails
- Contacts
- Photos
- Social Networks
- Bank Accounts
- Password Managers
- Access to corporate/internal servers
- Apps
- You name it...
# 1. APPS DEVELOPMENT PRICE

<table>
<thead>
<tr>
<th>App</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>$ 6000 – 50.00</td>
</tr>
<tr>
<td>Medium</td>
<td>$ 50.000 – 150.000</td>
</tr>
<tr>
<td>Complex</td>
<td>$ &gt; 150.0000</td>
</tr>
</tbody>
</table>

- [http://www.bluecloudsolutions.com/blog/cost-develop-app/](http://www.bluecloudsolutions.com/blog/cost-develop-app/)
1. Who are the apps developers?

Unmasking the App Developer

- 40% are 29 or younger
- 94% are men
- 2 years or less
- Amount of time that 49% have been an app developer
- 39% said app development was their full-time job
- 34% make less than $15,000 from app development
- 12% make $100,000 or more
- 40% work alone
- 71% are college graduates
- 27% work at 2-3 person firms

Source: GigaOm Pro web-based survey with 352 respondents, Sept. 2012
Graphic by Alberto Cervantes/The Wall Street Journal
## 1. APPS TECHNOLOGY

<table>
<thead>
<tr>
<th></th>
<th>iPhone</th>
<th>Android</th>
<th>Windows Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native</td>
<td>Objective-C</td>
<td>C (1)</td>
<td>C++ (WP8)</td>
</tr>
<tr>
<td>Managed</td>
<td></td>
<td>Java</td>
<td>NET</td>
</tr>
<tr>
<td>Web</td>
<td>Action Script, HTML5, CSS, JavaScript</td>
<td>Action Script, HTML5, CSS, JavaScript</td>
<td></td>
</tr>
<tr>
<td>Scripting</td>
<td>Ruby, Python</td>
<td>Python, Perl, JRuby, Lua, BeanShell, JavaSript, TCL and shell</td>
<td></td>
</tr>
<tr>
<td>3º Party / Free / Commercial</td>
<td>Java, C#</td>
<td>Visual Basic, C#</td>
<td>Java, HTML, CSS, JavaScript</td>
</tr>
</tbody>
</table>

(1) You can include C in Java Apps/Systems Apps written in C
1. WHY THE RISKS?

- Mobiles are computers

- 24x7x365 powered-on & Internet connected (always online)

- Used everywhere: airports, hotels, coffee shops, etc.

- Phone carriers responsible of updates
1. WHAT COULD GO WRONG? ATTACK VECTORS

- Contacts lost/scammed if phone lost/unlocked
- Fraudulent purchases via stored credit card/account details
- Malware apps/fake AV/bank apps
- Social media: reputational and malware harm
- QR code malware
- Geolocation via EXIF
- Scams/phishing 3x more successful
- Ransomware/harm to network security for organisation?
- Password/PIN/swipe usage?
- Location recovery app installed?
2. Write Secure Apps
2. COST OF FIXING BUGS

Source IDC and IBM Systems Sciences Institute
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
2. SECURE DEVELOPMENT FRAMEWORK

The Microsoft SDL

- Training
  - Core training

- Requirements
  - Analyze security and privacy risk
  - Define quality gates

- Design
  - Threat modeling
  - Attack surface analysis

- Implementation
  - Specify tools
  - Enforce banned functions
  - Static analysis

- Verification
  - Dynamic/Fuzz testing
  - Verify threat models/attack surface

- Release
  - Response plan
  - Final security review
  - Release archive

- Response
  - Response execution
2. THREAT MODELING (TM)

- Thinks to consider:
  - What App business purpose is
  - Who the user is
  - How is the App communicating with external servers, if any (At Transit)
  - What information is sent
  - How is information protected at device (At Rest)
  - Is crypto used? Is it secure enough?
  - Authentication/Authorization
  - What permissions we require
  - How resilient our App is, device jailbroken/rooted?
  - Do we validate data? How?
  - Do we logs errors? Where?
  - Etc...
2. SDL TM TOOL
2. APPS TO LEARN SECURE CODING & TESTING

- GoatDroid
  https://www.owasp.org/index.php/Projects/OWASP_GoatDroid_Project

- iGoat
  https://www.owasp.org/index.php/OWASP_iGoat_Project

- Exploit-Me Android
  https://github.com/SecurityCompass/AndroidLabs

- Exploit-Me iPhone
  https://github.com/SecurityCompass/iPhoneLabs

- Pandemobium
  https://github.com/denimgroup/Pandemobium
2. IOS MOBILE APPLICATION SECURITY

- Additional security controls for iOS against Apps attacks
  http://project-imas.github.io/index.html

- Provides
  - Apps Password
  - Encrypted Core Data
  - Secure Foundation
  - Security Check
  - Passcode Check
  - OpenSSL FIPS

- Free & Open Source!!
2. IMAS JAILBREAK DETECTION

```c
// Copyright (c) 2013 MITRE. All rights reserved.

#import "jailbreakCheckTemplate.h"

@interface jailbreakCheckTemplate :
@end

@implementation jailbreakCheckTemplate

-(id) init {
    self = [super init];
    if (nil != self) {
        
            // call wHereAProblem
            //
            cbBlock chkCallback = ^( weak id weakSelf = self;
                if (weakSelf) weakSelf wHereAProblem1;
            );
        
            // jailbreak detection
            //
            checkFork(chkCallback);
            checkFiles(chkCallback);
            checkLinks(chkCallback);

        return self;
    }
        
            // if the device is jailbroken then this method will be called
            //
            - (void) wHereAProblem {
                printf("We have a problem - jailbreak!");
            }
        }
```
2. ANDROID ROOTED DETECTION

- RootTools
  https://code.google.com/p/roottools/

```java
if (RootTools.isRootAvailable()) {
    // su exists, do something
} else {
    // do something else
}

if (RootTools.isAccessGiven()) {
    // your app has been granted root access
}
```
2. ANDROID ROOTED TESTER

**VULNEX ROOT TESTER**

<table>
<thead>
<tr>
<th>BASIC</th>
<th>ROOTTOOLS</th>
<th>ADVANCED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check One</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Two</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check Three</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Device is Rooted (2): Superuser.apk found**

**VULNEX ROOT TESTER**

<table>
<thead>
<tr>
<th>BASIC</th>
<th>ROOTTOOLS</th>
<th>ADVANCED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check BusyBox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check SuperUser</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check isbusybox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check isRootAvailable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check isAccessGiven</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check isNativeToolsReady</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check hasEnoughSpaceOnSdCard</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# 2. THEY ARE BACK! IOS

- Insecure API

<table>
<thead>
<tr>
<th>Insecure</th>
<th>Safer</th>
</tr>
</thead>
<tbody>
<tr>
<td>strcat</td>
<td>strlcat</td>
</tr>
<tr>
<td>strcpy</td>
<td>strlcpy</td>
</tr>
<tr>
<td>strncat</td>
<td>strlcat</td>
</tr>
<tr>
<td>strncpy</td>
<td>strlcpy</td>
</tr>
<tr>
<td>sprintf</td>
<td>Snprintf / asprintff</td>
</tr>
<tr>
<td>vsprintf</td>
<td>Vsnprintf / vasprintff</td>
</tr>
<tr>
<td>gets</td>
<td>fgets</td>
</tr>
</tbody>
</table>
2. HARDCODED CREDENTIALS

• Not too common but still possible to find Apps

```java
public final Connection getConnection() throws SQLException {
    return DriverManager.getConnection(
            "jdbc:mysql://localhost/dbName",
            "username", "password");
}
```

• DON’T!!!!
2. NOTHING LIKE REAL LIFE!

```bash
C:\Users\conde\Downloads\android-sdk_r07-windows\android-sdk-windows\tools>adb shell
# cd /data/data/martinicreations.passmanlite/
cd /data/data/martinicreations.passmanlite/
# ls
ls
lib
databases
shared_prefs
# cd shared_prefs
cd shared_prefs
# ls
ls
Passman.prefs.xml
# cat Passman.prefs.xml
cat Passman.prefs.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>
<boolean name="autoGenPassword" value="true" />
<int name="InactivityValue" value="150" />
</map>
```
2. STATIC ANALYSIS

• iOS
  – Clang Static Analyzer is your friend! http://amit-iphonedev.developer.blogspot.com

• Android
  – Agnitio http://sourceforge.net/projects/agnitio-tool/
2. ATTACK AT TRANSIT (I)

![Image of a login screen with the username 'jdoe' and a password '********']
2. ATTACK AT TRANSIT (II)
2. ATTACK AT TRANSIT (III)
2. FUZZ YOUR APPS
## 2. CRYPTO GUIDE

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>Banned (algorithms to be replaced)</th>
<th>Acceptable (for existing code, except sensitive data)</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric Bloc</td>
<td>DES, DESX, RC2, SKIPJACK</td>
<td>3DES (2 or 3 key)</td>
<td>AES (&gt;=128bit)</td>
</tr>
<tr>
<td>Symmetric Stream</td>
<td>SEAL, CYLINK_MEK, RC4 (&lt;128bit)</td>
<td>RC4 (&gt;= 128bit)</td>
<td>None, block cipher is preferred</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>RSA (&lt;2048bit), Diffie-Hellman</td>
<td>RSA (&gt;=2048bit), Diffie-Hellman (&gt;=2048bit)</td>
<td>RSA (&gt;=2048bit), Diffie-Hellman</td>
</tr>
<tr>
<td></td>
<td>(&lt;2048bit)</td>
<td></td>
<td>(&gt;=2048bit), ECC (&gt;=256bit)</td>
</tr>
<tr>
<td>Hash (includes HAMC)</td>
<td>SHA-0, SHA-1, MD2, MD4, MD5</td>
<td>SHA-2</td>
<td>SHA-2 (includes: SHA-256, SHA-384, SHA-512)</td>
</tr>
<tr>
<td>HAMC Key Lengths</td>
<td>&lt;112bit</td>
<td>&gt;=112bit</td>
<td>&gt;=128bit</td>
</tr>
</tbody>
</table>
2. WATCH OUT FOR THE CLASSICS!

• Some attacks against your App:
  1. Cross-Site Scripting (XSS)
  2. SQL Injection (SQLi)
  3. Session Tampering
  4. Verbose Error Messages
  5. Insecure Channels
  6. Improper Data Validation
  7. Buffer Overflows
2. FREE SECURITY DEVELOPMENT TRAINING

- SAFECODE
  https://training.safecode.org/courses

- Microsoft SDL
2. NO SEC SKILLS IN THE HOUSE? NO PROBLEM

• Bug Hunting Programs to the rescue!

• We can create a program in-house or use any of the bug hunting companies
2. MEDIA MONITORING

• After App released need to do monitoring/ OSINT:
  – Is our App being cracked?
  – Are we on XCon list (iPhone)?
  – Any bug reported?

• Keep an eye on:
  – Social Networks
  – Forums
3. Conclusions
3. SUMMARY

• Apps:
  – are serious money!
  – are being developed like the Web
3. MOBILE SECURITY, TIME FOR CHANGE!

- Apps need to pass Software Security Assurance practices
- Threat Model your Apps
- Understand platform and Apps risks
3. RESOURCES

• OWASP Mobile Project
  https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Project_Overview

• Mobile Application Development

• Veracode
  https://www.veracode.com/blog/2013/08/developers-guide-to-building-secure-mobile-applications-infographic/
3. Q&A

- Thanks!

- Twitter: @simonroses / @vulnexsl

- www.vulnex.com

- www.simonroses.com