

---

# SMARTPHONE APPS ARE NOT THAT SMART

---

# ME?

---

VULNEX: [www.vulnexus.com](http://www.vulnexus.com)

Blog: [www.simonroses.com](http://www.simonroses.com)

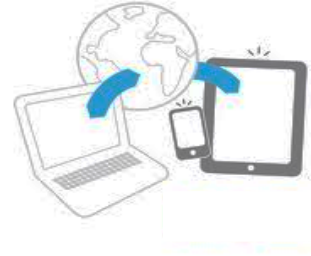
Twitter: @simonroses

VULNEX

# TALK OBJECTIVES

---

- Apps are the new Web



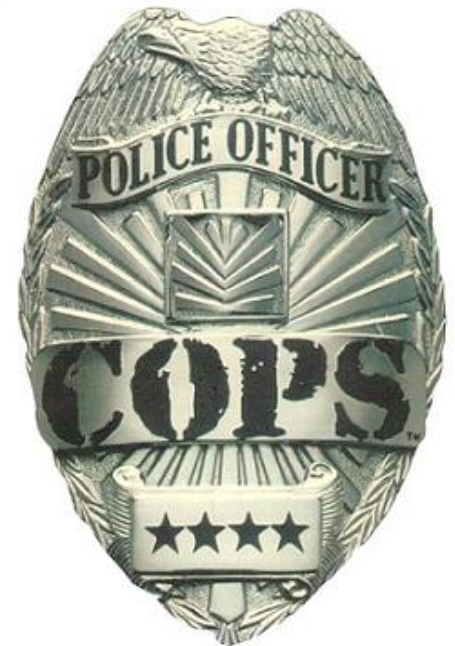
- Peek into current state of Apps security on Markets
- Bugs will be revealed but not the victims



# DISCLAIMER

---

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



**VULNEX**



# 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?

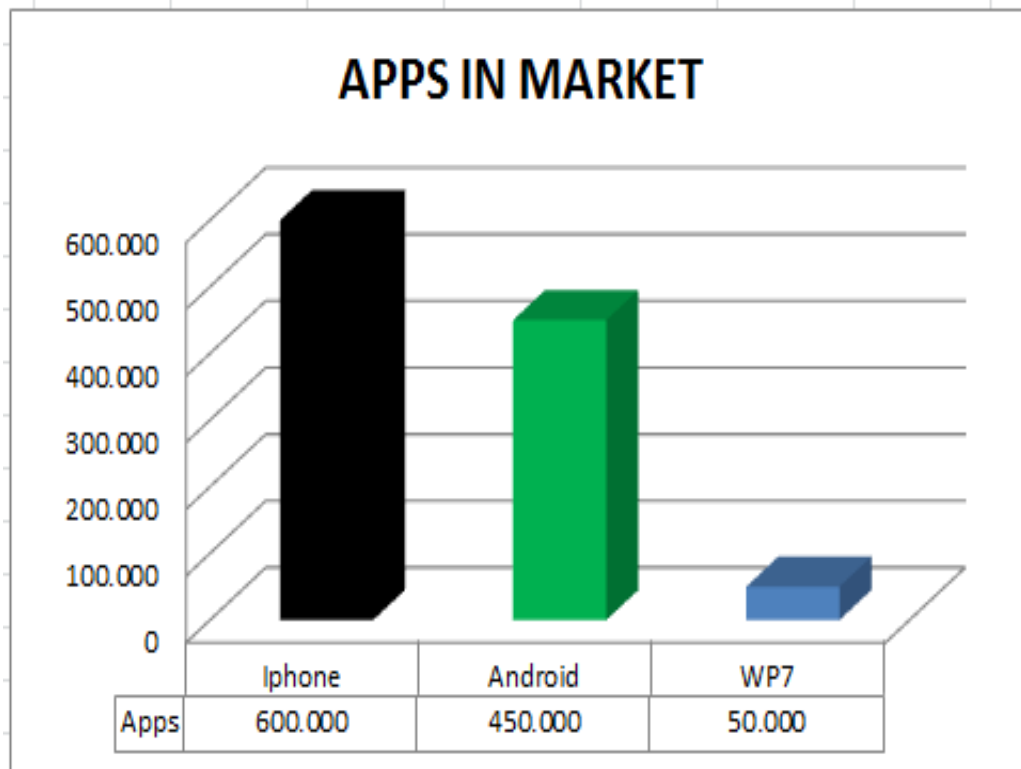
---

- 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: ¿?



**VULNEX**

# 1. SMARTPHONE DEVELOPMENT

---

	IPhone	Android	WP7
Managed		Java	.NET
Native	Objective-C	C <sub>(1)</sub>	(2)
Web	Action Script, HTML, CSS, JavaScript	Action Script, HTML, CSS, JavaScript	
Scripting	Ruby	Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl, and shell	
3° Party / Free / Commercial	Java, c#	Visual Basic, c#	Java, HTML, CSS, JavaScript

(1) Parts of C in Java Apps / Full C apps at platform level

(2) 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

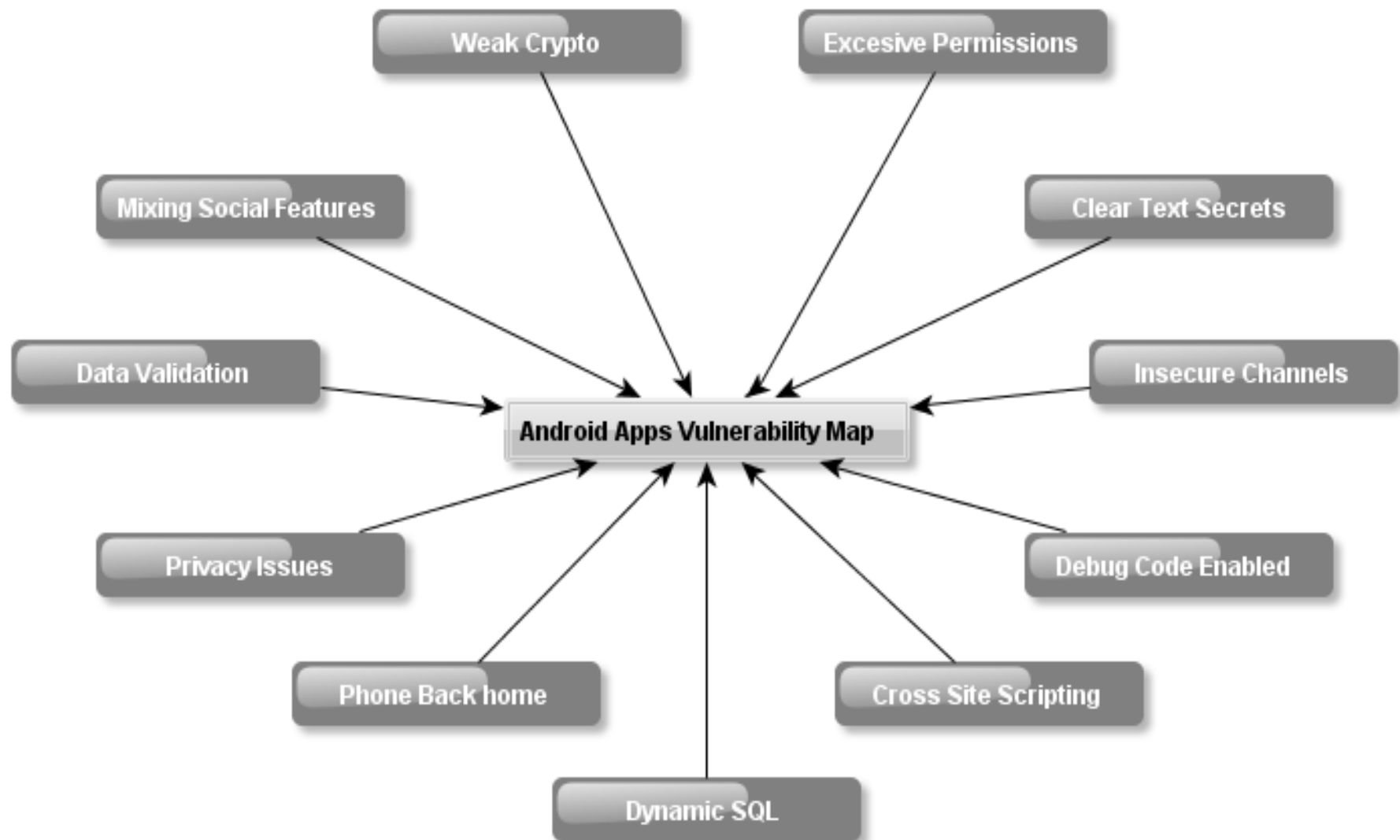
## 2. OWASP MOBILE TOP 10 RISKS

---

- M1 • Insecure Data Storage
- M2 • Weak Server Side Controls
- M3 • Insufficient Transport Layer Protection
- M4 • Client Side Injection
- M5 • Poor Authorization and Authentication
- M6 • Improper Session Handling
- M7 • Security Decisions Via Untrusted Inputs
- M8 • Side Channel Data Leakage
- M9 • Broken Cryptography
- M10 • Sensitive Information Disclosure

## 2. VULNEX APPS RISKS

---



---

## **3. CASE STUDIES**

---

### 3. CLEAR TEXT SECRETS

---

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

Data Storages		
IPhone	Android	WP7
Core Data	Shared Preferences	Isolated Storage
SQLite Databases	Internal Storage	Network Connection
Logs	External Storage	
Network Connection	SQLite Databases	
Property List (plist)	Network Connection	
XML		

### 3. CLEAR TEXT SECRETS EXAMPLE: CREDENTIALS MANAGER (CVE-2011-1840)

```
C:\Users\conde\Downloads\android-sdk_r07-windows\android-sdk-windows\tools>adb s
hell
# 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="HideSensitiveData" value="true" />
<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://[REDACTED].com/api/ HTTP/1.1

Host: [REDACTED].com

User-Agent: [REDACTED]

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": "3c2bc9cdd5a9d5ae5437aea5a8df3a36", "seed": "ABsXShfLj1TQ2iZmM+4Kte6og4l18exmD4SU6cPgwDg=", "email": "test", "timestamp": "1330638622", "application_key": "MDI3MDFmZjU4MGExNWYmYjA5MzRkODImMjg0MTU6MC43NzQ4ODAwMCAxMjc1NDcyNjgz" } ], [ "getVersion", {} ] ] }
```

### 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



```
DataSource dataSource = ...  
Connection connection = dataSource.getConnection();  
Statement statement = connection.createStatement();  
ResultSet resultSet = statement.executeQuery("SELECT * FROM ...");  
while(resultSet.next()) {  
    // ...  
}
```

- Helps attacker to learn Apps internal
- OWASP Mobile: M8- Side Channel Data Leakage



### 3. DEBUG ENABLED EXAMPLE: FINANCE

---

```
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 version="1.0" encoding="UTF-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.server" 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.server.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.server.LoginActivity"/>
    <service android:name="com.example.server.ServerService"/>
    <activity android:name="com.google.ads.AdActivity" android:configChanges="keyboard|keyboardHidden|orientation"/>
  </application>
  <uses-sdk android:minSdkVersion="4"/>
</manifest>
```

### 3. DEBUG ENABLED EXAMPLE: FINANCE

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text: to limit scope. verbose    

L...	Time	PID	Application	Tag	Text
I	03-01 23:28:3...	452		jdwp	received file descriptor 10 from ADB
D	03-01 23:28:3...	452		ddm-heap	Got feature list request
D	03-01 23:28:3...	452	com.b...	dalvikvm	GC freed 2765 objects / 220768 bytes in 84ms
D	03-01 23:28:3...	452	com.b...	TermsActivity	onCreate
D	03-01 23:28:3...	452	com.b...	loadTermsInfo	loadTermsInfo
D	03-01 23:28:3...	452	com.b...	loadTermsInfo	StoredFileName->/[REDACTED]...
D	03-01 23:28:3...	452	com.b...	loadTermsInfo	StoredTimeStamp->634384532369648955
D	03-01 23:28:3...	452	com.b...	TermsActivity	onResume
D	03-01 23:28:3...	452	com.b...	HttpClient	get url: http://[REDACTED]...
D	03-01 23:28:3...	452	com.b...	HttpClient	OpenHttpConnection http://[REDACTED]...
D	03-01 23:28:3...	452	com.b...	HttpClient	POST contents: platform=[REDACTED]&app_country=[REDACTED]&app_version=[REDACTED]&inter...
W	03-01 23:28:3...	452	com.b...	ExpatReader	DTD handlers aren't supported.
D	03-01 23:28:3...	452	com.b...	(requestOp...	RequestOperation successful.
D	03-01 23:28:3...	452	com.b...		loadTermsInformationDownloaded

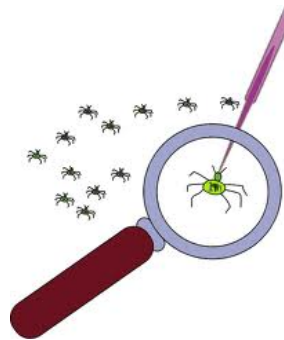
### 3. DEBUG ENABLED MITIGATION

---

- 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



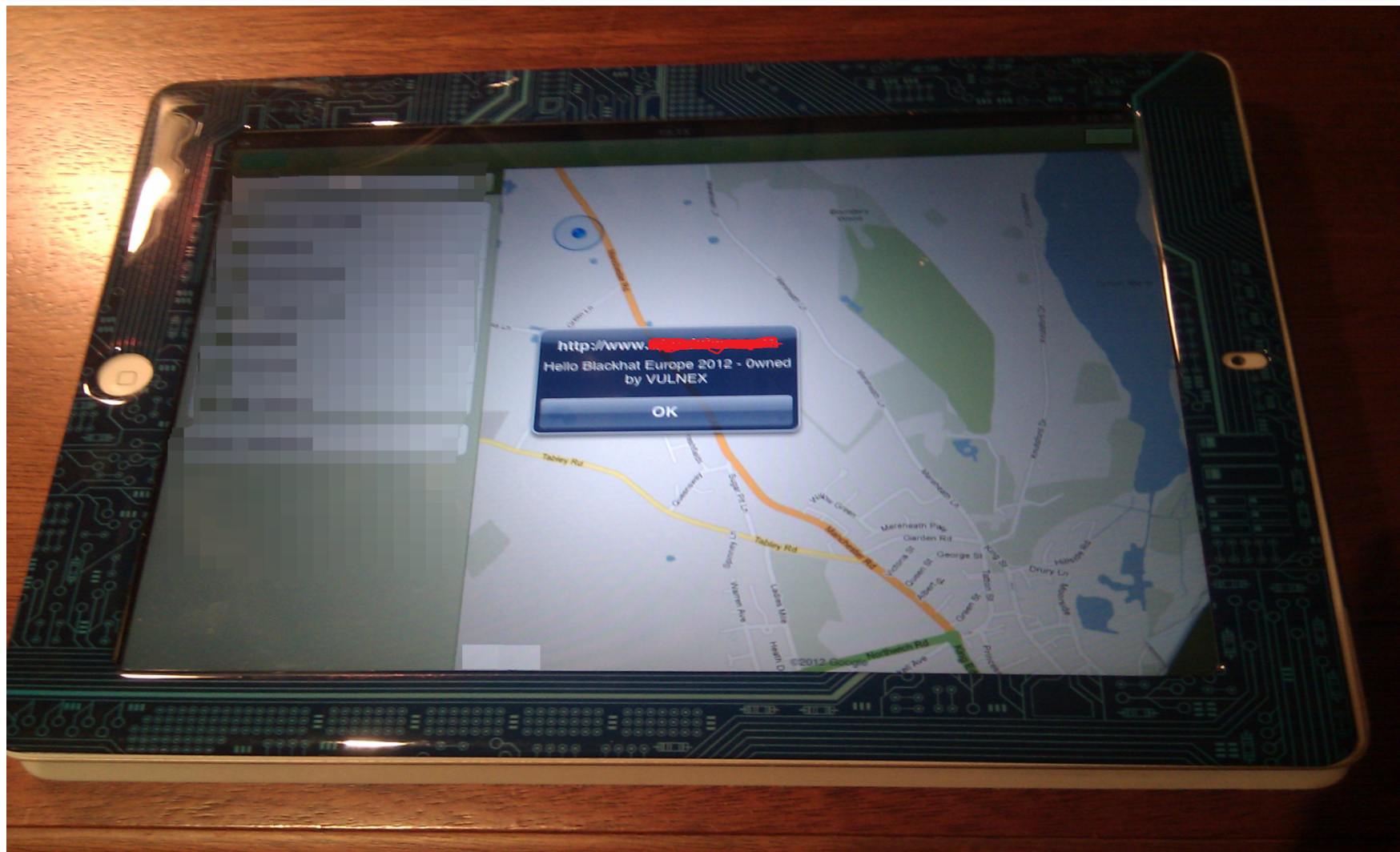
### 3. DYNAMIC SQL EXAMPLE: FINANCE

---

```
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:

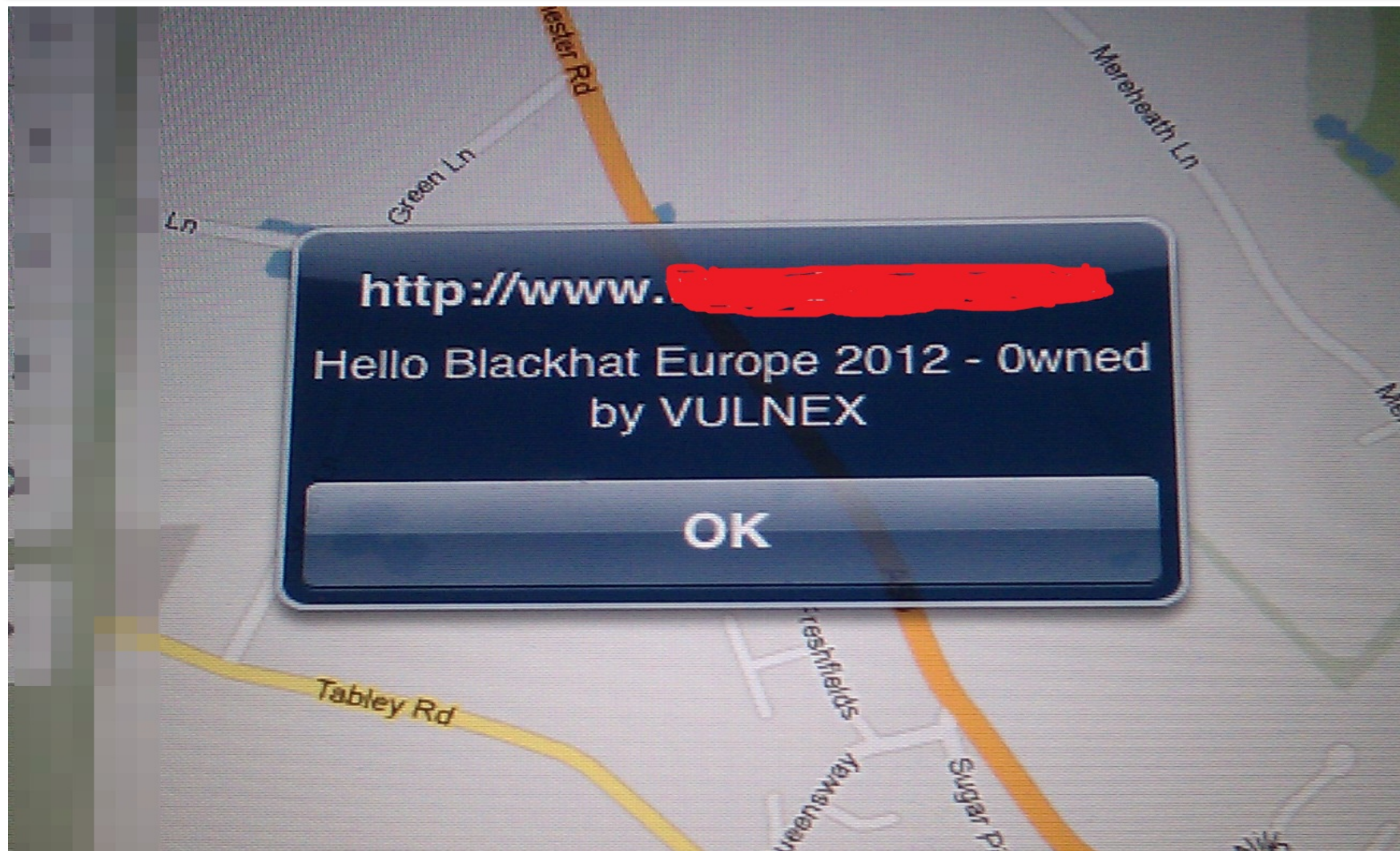
---





### 3. CROSS SITE SCRIPTING (XSS) EXAMPLE, IN CASE YOU MISSED IT

---



### 3. DATA VALIDATION EXAMPLE: MEDIA

```
172.17.17.64 - PuTTY
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "--host=arm-apple-darwin9 --target=..."
Reading symbols for shared libraries .. done

(gdb) r
Starting program: /private/var/mobile/Applications/[REDACTED]
[REDACTED]
[REDACTED]
2012-03-01 22:14:27.570 [437:3f0f] displayInfoUpdate:
2012-03-01 22:14:27.583 [437:303] Started HTTP server on port 49345
2012-03-01 22:14:27.607 [437:303] Listening on 2121
2012-03-01 22:14:27.610 [437:303] displayInfoUpdate:
2012-03-01 22:14:28.506 [437:303] Bonjour Service Published: domain(local.) type(http_tcp.) name([REDACTED])
2012-03-01 22:14:48.754 [437:303] Incorrect NSStringEncoding value 0x0000 detected. Assuming NSASCIIStringEncoding. Will stop this compatibility m
apping behavior in the near future.
2012-03-01 22:14:48.759 [437:303] FC: Current Directory starting at /private/var/root/Library/Caches/TempAVFiles
2012-03-01 22:14:48.763 [437:303] FS:didAcceptNewSocket port:2121
2012-03-01 22:14:48.767 [437:303] Connection on Server Port
2012-03-01 22:14:48.774 [437:303] FC:onSocketWillConnect
2012-03-01 22:14:51.203 [437:303] <USER anon
2012-03-01 22:14:51.207 [437:303] >331 Password required for anon
2012-03-01 22:14:53.025 [437:303] <PASS anon
2012-03-01 22:14:53.029 [437:303] >230 User anon logged in.
2012-03-01 22:14:56.733 [437:303] <CWD [REDACTED]
2012-03-01 22:14:56.737 [437:303] DoCwd arguments is (
    CWD,
    [REDACTED]
)
2012-03-01 22:14:56.742 [437:303] (
    CWD,
    [REDACTED]
)
2012-03-01 22:14:56.746 [437:303] HYPHEN FOUND IGNORE
2012-03-01 22:14:56.750 [437:303] *** Terminating app due to uncaught exception 'NSRangeException', reason: '-[__NSCFConstantString characterAtIn
dex:]: Range or index out of bounds'
*** First throw call stack:
(0x336e78bf 0x301db1e5 0x336e77b9 0x336e77db 0x3365501d 0x7d447 0x7d5b1 0x7e8e5 0x33641435 0x7f8b1 0x7ecd1 0x7dd0d 0x38e3b 0x3c965 0x3c527 0x385af 0x3367c563
 0x3367c511 0x3367c383 0x338be119 0x3384ac69 0x3384abe9 0x336be31f 0x336bbb03 0x336bb2cf 0x336ba075 0x3363d4dd 0x3363d3a5 0x37daffcd 0x310a0743 0x72c5 0x725c
)
terminate called throwing an exception
Program received signal SIGABRT, Aborted.
0x3521232c in ?? ()
(gdb)
```

### 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?



**VULNEX**

### 3. PERMISSIONS

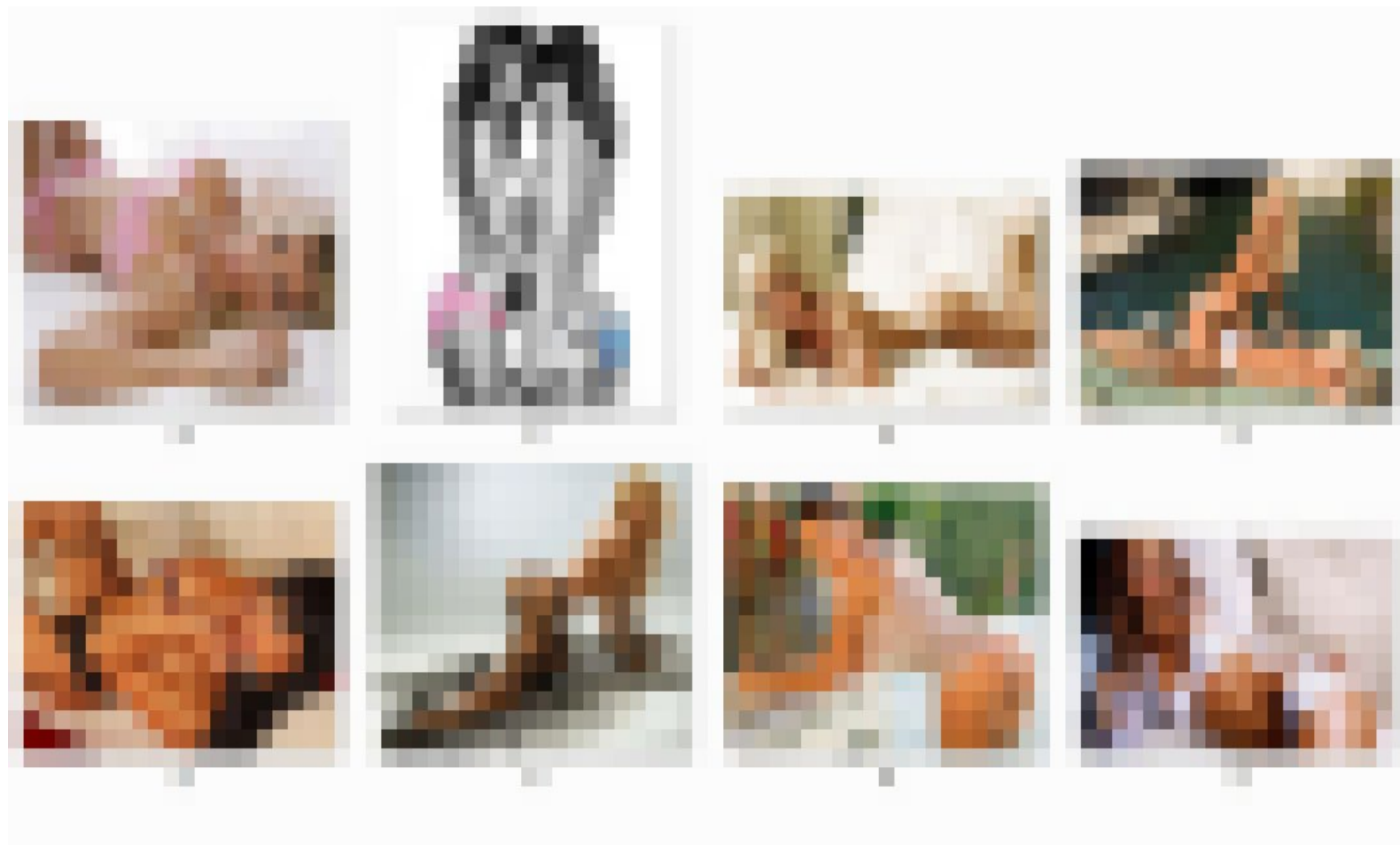
---

- It's important to understand App permissions
- App can compromise device security and user pocket



### 3. PERMISSIONS EXAMPLE - SEXYPIC

---



# 3. PERMISSIONS EXAMPLE - DROIDDREAM

Electric Sheep [F:\apks\malware\com.space.sexyipic] V 0.1

File Tools Help

res  
smali  
com  
space  
sexyipic  
image.smali  
PicViewer\$1.sma  
PicViewer\$2.sma  
PicViewer.smali  
Pusher\$1.sma  
Pusher.smali  
R\$attr.smali  
R\$drawable.smali  
R\$id.smali  
R\$layout.smali  
R\$string.smali  
R\$styleable.smali  
R.smali  
SexyPic\$1\$1.sma  
SexyPic\$1.smali  
SexyPic\$2.smali  
SexyPic.smali  
SmsReceiver.sma  
Utils.smali  
AndroidManifest.xml  
apktool.yml

F:\apks\malware\com.space.sexyipic\AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest android:versionCode="1" android:versionName="1.0" package="com.space.sexyipic"
3 xmlns:android="http://schemas.android.com/apk/res/android">
4 <uses-permission android:name="android.permission.INTERNET" />
5 <uses-permission android:name="android.permission.READ_PHONE_STATE" />
6 <uses-permission android:name="android.permission.READ_SMS" />
7 <uses-permission android:name="android.permission.RECEIVE_SMS" />
8 <uses-permission android:name="android.permission.SEND_SMS" />
9 <application android:label="@string/app_name" android:icon="@drawable/icon" android:debuggable="true">
10 <activity android:label="@string/app_name" android:name=".SexyPic" android:screenOrientation="portrait" android:configChanges="keyboardHidden|orientation">
11 <intent-filter>
12 <action android:name="android.intent.action.MAIN" />
13 <category android:name="android.intent.category.LAUNCHER" />
14 </intent-filter>
15 </activity>
16 <activity android:name=".PicViewer" android:screenOrientation="portrait" android:configChanges="keyboardHidden|orientation" />
17 <receiver android:name=".SmsReceiver" android:enabled="true">
18 <intent-filter android:priority="100">
19 <action android:name="android.provider.Telephony.SMS_RECEIVED" />
20 </intent-filter>
21 </receiver>
22 <service android:name=".Pusher" />
23 </application>
24 </manifest>
25
```

App Permissions (5)

Name	Risk	Desc
android.permission.INTERNET	Yellow	API Level 1 Allows applications to open network sockets.
android.permission.READ_PHONE_STATE	Yellow	API Level 1 Allows read only access to phone state.
android.permission.READ_SMS	Yellow	API Level 1 Allows an application to read SMS messages.
android.permission.RECEIVE_SMS	Yellow	API Level 1 Allows an application to monitor incoming SMS messages, to record or perform processing on them.
android.permission.SEND_SMS	Red	API Level 1 Allows an application to send SMS messages.

Please, select target to reverse...

### 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

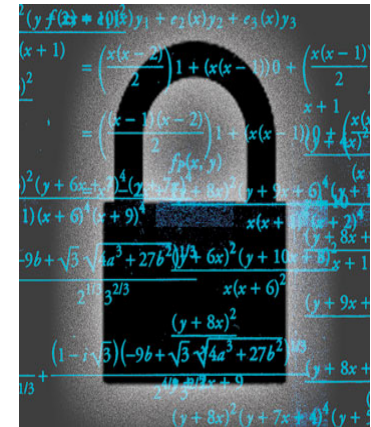
---

Default password in code

Encrypt pwd with MD5 (no salt)

Pwd stored in text file with world perms

File stored in SD card



**MD5 hash**  
huge online hash database

<http://www.md5-hash.com/>

**VULNEX**

### 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

---

```
private static final String LOGIN_URL = "[REDACTED]";  
private static final String PASSWORD = "tetra1404ad3772";  
public static final String PREFS_NAME = "[REDACTED]";
```

### 3. HARDCODED 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 to directly to /data/data
  - iPhone
    - Call fork()
    - Check if /Applications/Cydia.app exist
  - WP7
    - Allowed by Microsoft, <http://labs.chevronwp7.com/>
- Code Obfuscation



## 4. SECURITY RESOURCES

---

- Iphone
  - <https://developer.apple.com/library/mac/#documentation/security/Conceptual/SecureCodingGuide/Introduction.html>
- Android
  - <http://developer.android.com/guide/topics/security/security.html>
  - <http://developer.android.com/search.html?q=security&t=5>
- WP7
  - [http://msdn.microsoft.com/en-us/library/ff402533\(VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402533(VS.92).aspx)
- OWASP Mobile Security Project  
[https://www.owasp.org/index.php/OWASP\\_Mobile\\_Security\\_Project](https://www.owasp.org/index.php/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



Tool that provides  
information about  
the state of  
software  
code  
during  
execution.



- Study cross platform technologies and their impact on security
  - Managed Apps (Mono)
  - Are bug cross platform?



**VULNEX**

## 5. Q&A

---

- Please fill out the black hat feedback form
- Thanks!