



# Security and Privacy issues in iOS and Android Apps

**OWASP**  
July 12, 2011

Praveen Nallasamy  
OWASP Leader, NYC Chapter  
[Praveen.nallasamy@owasp.org](mailto:Praveen.nallasamy@owasp.org)  
[Praveen.nallasamy@gmail.com](mailto:Praveen.nallasamy@gmail.com)  
[www.praveennallasamy.com](http://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.

**The OWASP Foundation**  
<http://www.owasp.org>

# Agenda

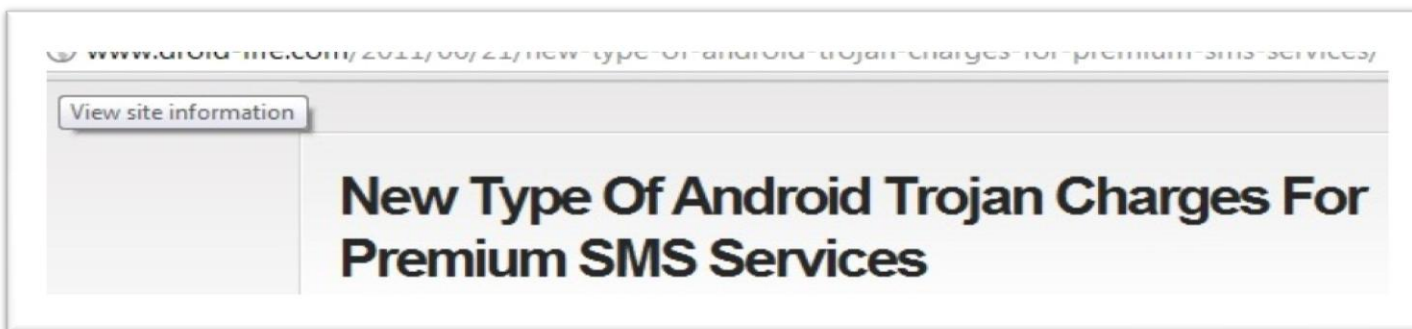
- About This Talk
- Need for Mobile Security
- Top 5 Issues
  1. Device Tracking
  2. Insecure Storage
  3. Insecure Communication
  4. Excessive Permissions
  5. Web Based vulnerabilities
- Questions
  - ▶ Love to answer all of them at the end of the presentation

# 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

TOP 5

# 1. Device Tracking

## What are Device Identifiers ?

Think of them as similar to the VIN number of a vehicle.

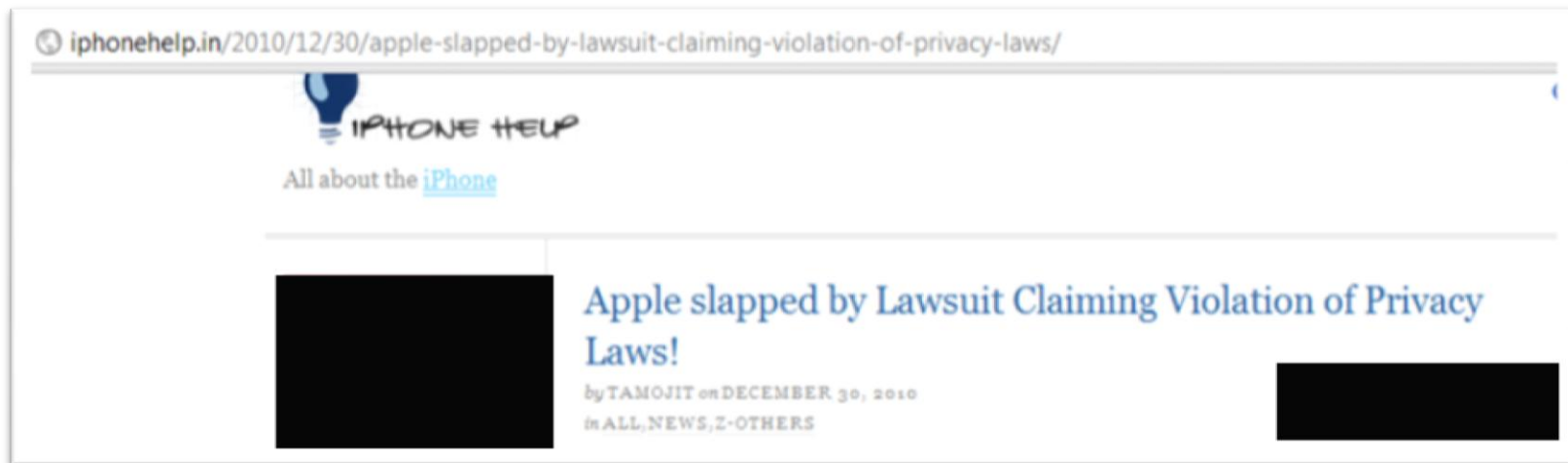
1. UDID (Unique Device Identifier) – Apple Serial Number
  2. IMEI ( International Mobile Equipment Identity) Number – Unique GSM number, applicable to Android and iOS as long as it is on a GSM phone
- Most Apps collect at least one of the device identifiers
  - App owners collect them
  - Third party 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.
  - ▶ Recent law suits have surfaced against Apple, Weather Channel and Pandora for collecting UDIDs.

<http://iphonehelp.in/2010/12/30/apple-slapped-by-lawsuit-claiming-violation-of-privacy-laws/>





# 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 boundries](http://digitalarchive.gsu.edu/cgi/viewcontent.cgi?article=1850&context=gsulr&sei-redir=1#search=)

## GEORGIA STATE UNIVERSITY LAW REVIEW

---

---

VOLUME 17

NUMBER 2

WINTER 2000

---

---

### CHIPPING AWAY AT THE BOUNDARIES OF PRIVACY: INTEL'S PENTIUM III PROCESSOR SERIAL NUMBER AND THE EROSION OF FOURTH AMENDMENT PRIVACY EXPECTATIONS

George M. Dery III<sup>†</sup>

James R. Fox<sup>††</sup>



# 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.

www.numberingplans.com/?page=analysis&sub=imei#r



## INTERNATIONAL numbering plans

**Services**

**Subscriptions**

**Numbering plans**

**Number analysis tools**

**On-line dialling tools**


**Databases**

**Contact**

**Analysis of IMEI numbers**

All mobile phones are assigned a unique 15 digit IMEI code upon production. Below you can check all known information regarding manufacturer, model type, and country of approval of a handset.


**Tip!** The IMEI can be displayed on most mobile handsets by dialling \*#06#. Otherwise check the compliance plate under the battery.



**Enter IMEI number below**

*Example: 350077-52-323751-3*

**Information on IMEI** [redacted]

Type Allocation Holder	HTC
Mobile Equipment Type	HTC Google Nexus One
GSM Implementation Phase	2/2+
IMEI Validity Assessment	 Very likely

**Information on range assignment**

Est. Date of Range Issuance	Around Q2 2009
Reporting Body	British Approvals Board of Telecommunications (BABT)
Primary Market	Europe
Legal Basis for Allocation	EU R&TTE Directive

**Information on number format**

Full IMEI Presentation	[redacted]
Reporting Body Identifier	35
Type Allocation Code	[redacted]
Serial Number	[redacted]
Check Digit	8

**Your account**

E-mail address

Password

Log in

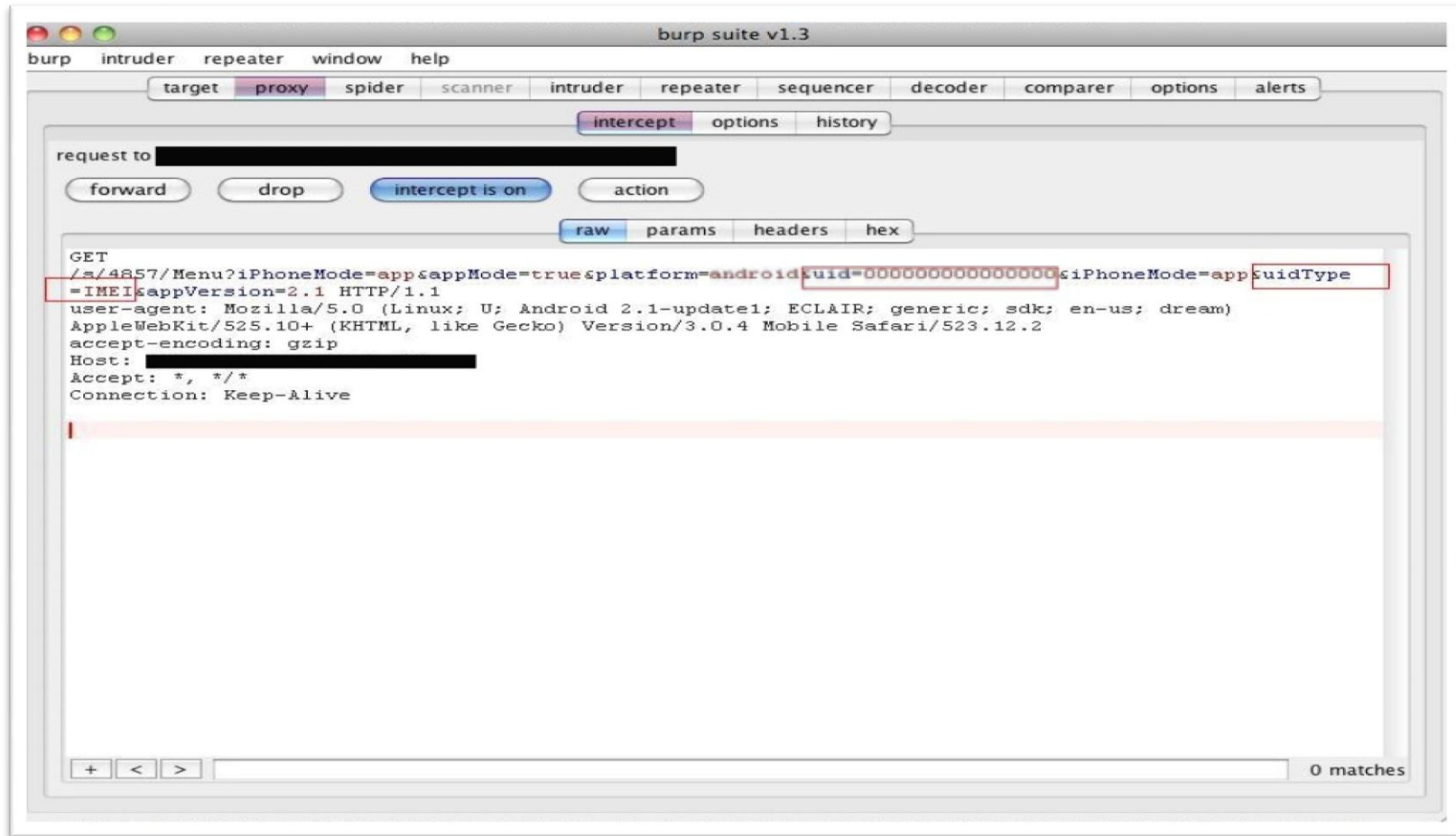
Neustar UltraDNS  
Premium DNS  
Managed Service  
Used by the  
Biggest Internet  
Companies!  
www.UltraDNS.com

Ads by Google



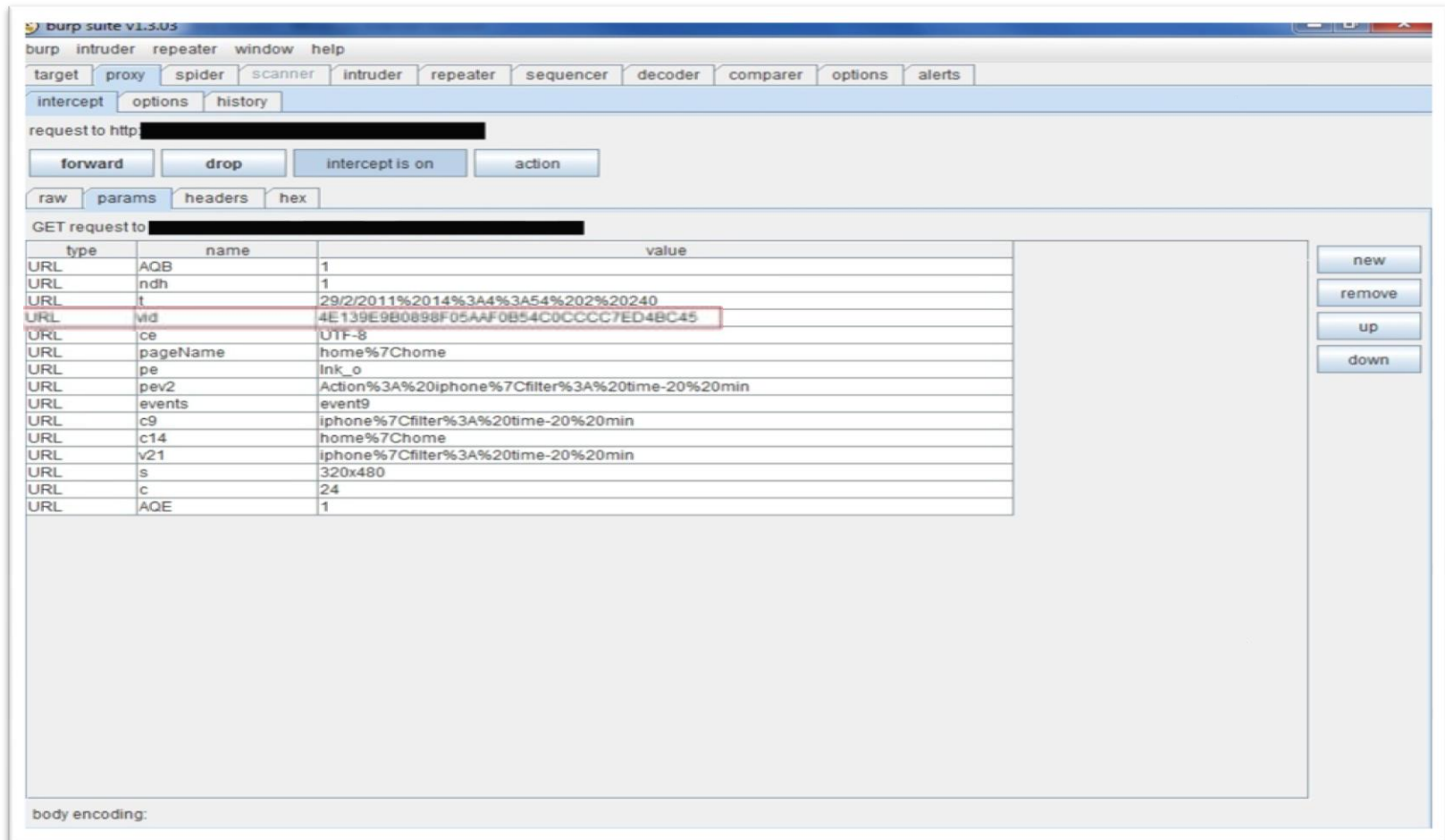
# 1. Device Tracking

Example: Android app transmitting IMEI number in clear text.



# 1. Device Tracking

Example: iOS app transmitting UDID in clear text.



The screenshot shows the Burp Suite interface with a GET request intercepted. The 'params' tab is selected, displaying a table of request parameters. The 'vid' parameter is highlighted with a red box, showing its value as a hexadecimal string representing an iOS UDID.

type	name	value
URL	AQB	1
URL	ndh	1
URL	t	29/2/2011%2014%3A4%3A54%20%20240
URL	vid	4E139E9B0898F05AAF0B54C0CCCC7ED4BC45
URL	ce	UTF-8
URL	pageName	home%7Chome
URL	pe	lnk_o
URL	pev2	Action%3A%20iphone%7Cfilter%3A%20time-20%20min
URL	events	event9
URL	c9	iphone%7Cfilter%3A%20time-20%20min
URL	c14	home%7Chome
URL	v21	iphone%7Cfilter%3A%20time-20%20min
URL	s	320x480
URL	c	24
URL	AQE	1

# 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:

```
# pwd
pwd
/data/data
# ls
ls
com.android.providers.contacts
com.android.launcher
com.android.alarmclock
com.android.mms
com.android.providers.downloads
com.android.providers.applications
com.android.quicksearchbox
com.android.providers.telephony
com.android.providers.media
com.android.email
com.android.browser
com.android.spare_parts
```

## 2. Insecure data storage

Example: Android app storing device ID in plain text in a XML file.

```
cal
at
?xml version='1.0' encoding='utf-8' standalone='yes' ?>
map>
string name="totalSessionTime">0</string>
string name="AnalyticsServer">http://                               /string>
string name="UserAgent">Mozilla/5.0 (Linux; U; Android 2.2; en-us; sdk Build/FRF91) AppleWebKit/533
1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1</string>
string name="totalIdleTime">0</string>
string name="offlineSessions">0</string>
string name="ConfigServer"                                     </string>
string name="pollTime">30000</string>
string name="UcServer"                                       </string>
string name="320x50">true,60000,false</string>
string name="AdServer">http                                     </string>
string name="idleTimeout">120000</string>
string name="300x250">true,60000,false</string>
string name="FeedbackServer"                                   /string>
string name="deviceId">0000000000000000</string>
/map>
```



## 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`

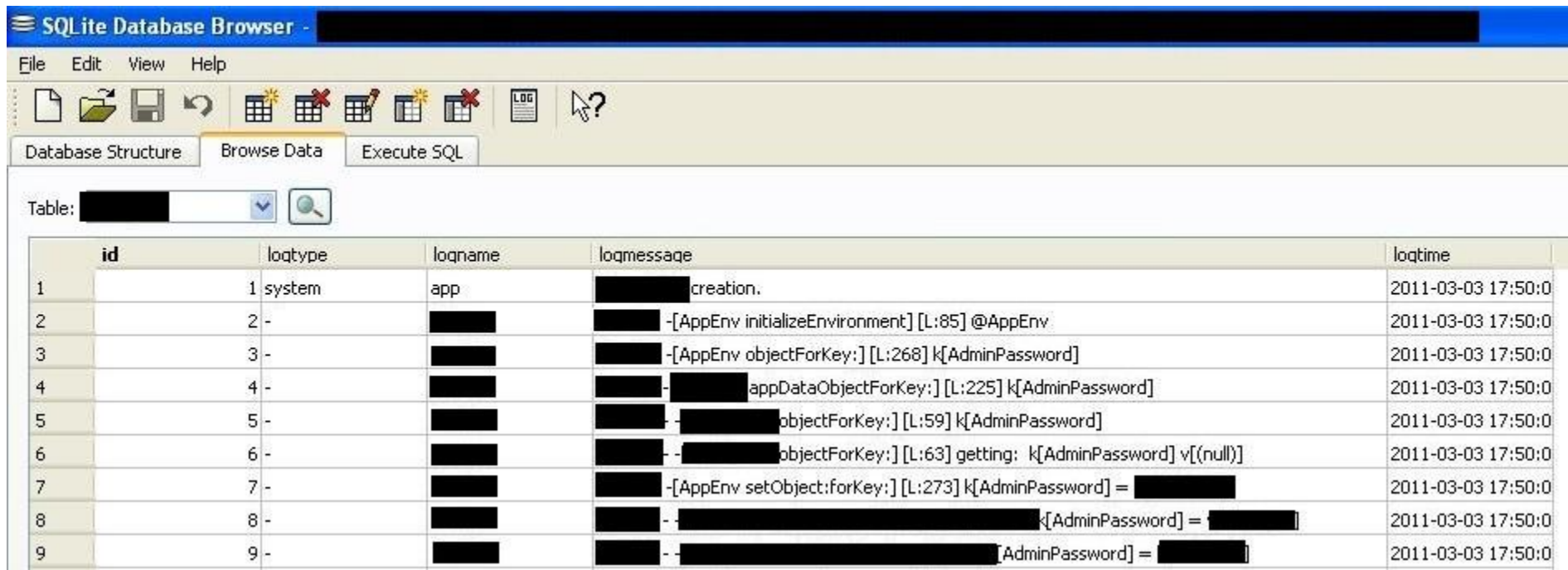
## 2. Insecure data storage

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.

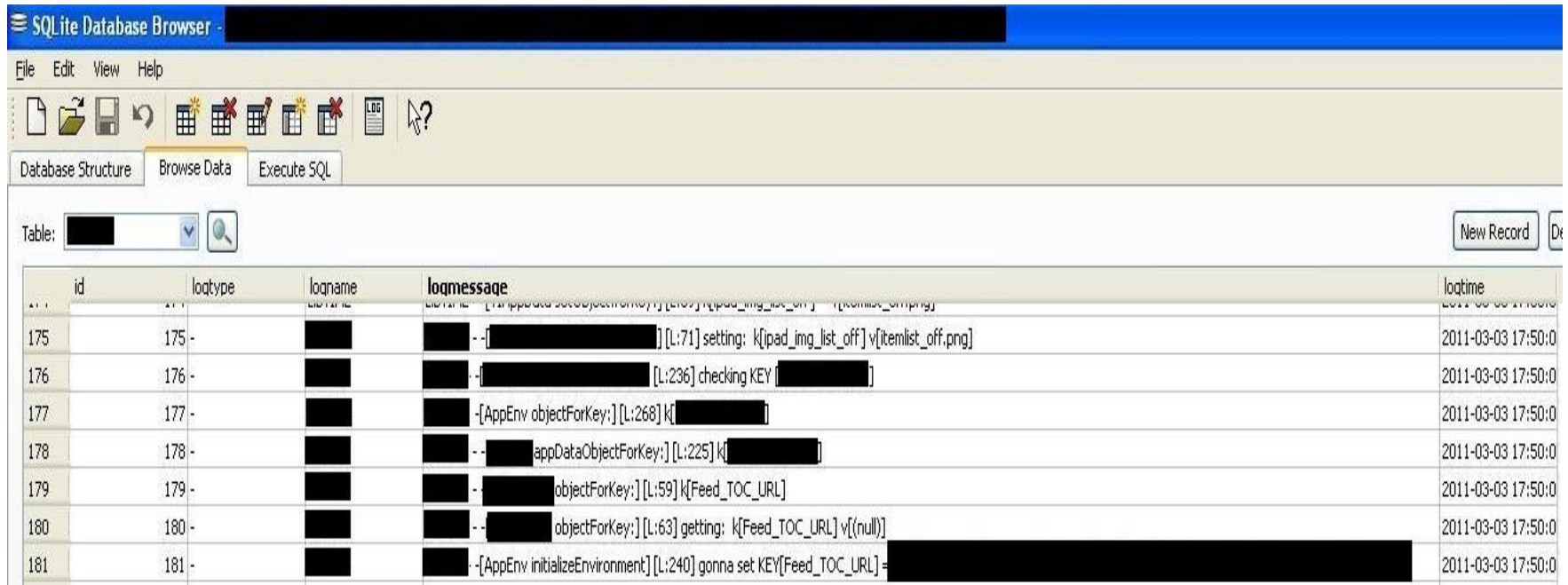


The screenshot shows the SQLite Database Browser interface. The table selected is named [REDACTED]. The table contains 9 rows of log data. The columns are id, loctype, loqname, loqmessage, and loqtime. The log messages show the app's initialization and several calls to k[AdminPassword], indicating that the admin password is stored in clear text.

id	loctype	loqname	loqmessage	loqtime
1	1 system	app	[REDACTED] creation.	2011-03-03 17:50:0
2	2 -	[REDACTED]	[REDACTED] -[AppEnv initializeEnvironment] [L:85] @AppEnv	2011-03-03 17:50:0
3	3 -	[REDACTED]	[REDACTED] -[AppEnv objectForKey:] [L:268] k[AdminPassword]	2011-03-03 17:50:0
4	4 -	[REDACTED]	[REDACTED] -[AppEnv objectForKey:] [L:225] k[AdminPassword]	2011-03-03 17:50:0
5	5 -	[REDACTED]	[REDACTED] -[AppEnv objectForKey:] [L:59] k[AdminPassword]	2011-03-03 17:50:0
6	6 -	[REDACTED]	[REDACTED] -[AppEnv objectForKey:] [L:63] getting: k[AdminPassword] v[(null)]	2011-03-03 17:50:0
7	7 -	[REDACTED]	[REDACTED] -[AppEnv setObject:forKey:] [L:273] k[AdminPassword] = [REDACTED]	2011-03-03 17:50:0
8	8 -	[REDACTED]	[REDACTED] -[AppEnv setObject:forKey:] [L:273] k[AdminPassword] = [REDACTED]	2011-03-03 17:50:0
9	9 -	[REDACTED]	[REDACTED] -[AppEnv setObject:forKey:] [L:273] k[AdminPassword] = [REDACTED]	2011-03-03 17:50:0

# 2. Insecure data storage

Example: iOS app storing Active Directory passwords.

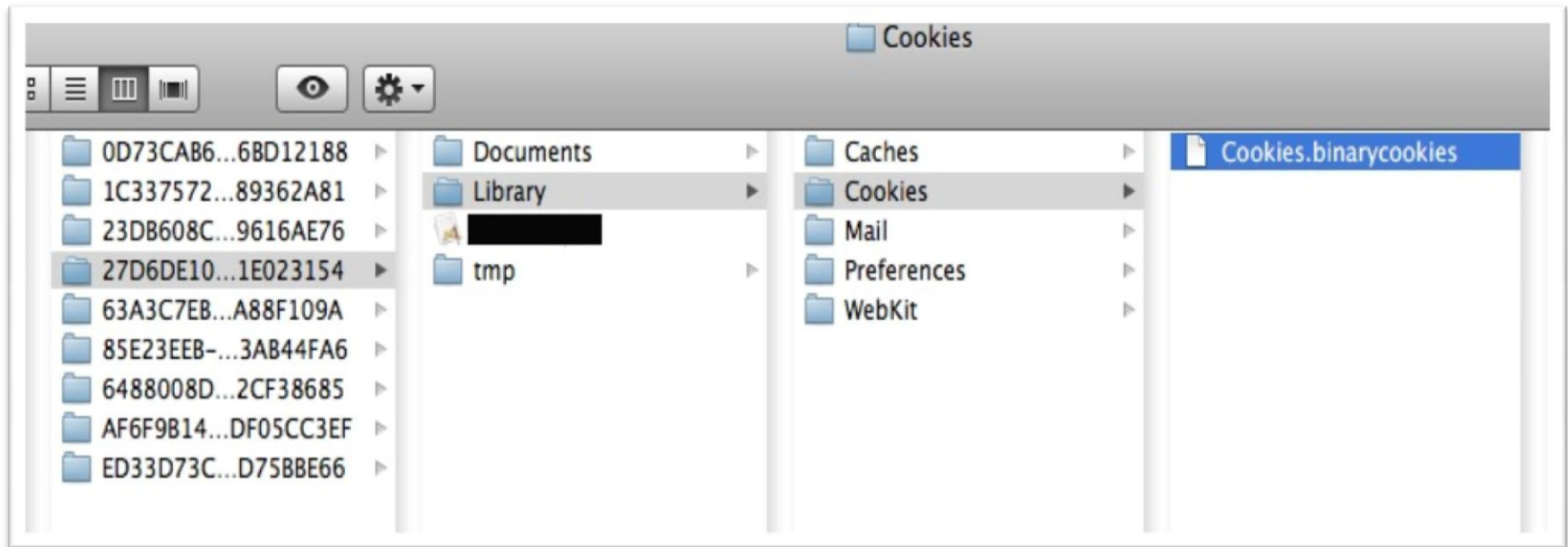


The screenshot shows the SQLite Database Browser interface. The table 'logmessage' contains several rows of log data. The 'logmessage' column contains sensitive information, including Active Directory passwords and other system-related data. The 'logtime' column shows the timestamp for each log entry.

id	logtype	logname	logmessage	logtime
175	175	[REDACTED]	[REDACTED] [L:71] setting: k[ipad_img_list_off] v[itemlist_off.png]	2011-03-03 17:50:0
176	176	[REDACTED]	[REDACTED] [L:236] checking KEY [REDACTED]	2011-03-03 17:50:0
177	177	[REDACTED]	[REDACTED] [AppEnv objectForKey:] [L:268] k[REDACTED]	2011-03-03 17:50:0
178	178	[REDACTED]	[REDACTED] [AppDataObjectForKey:] [L:225] k[REDACTED]	2011-03-03 17:50:0
179	179	[REDACTED]	[REDACTED] objectForKey:] [L:59] k[Feed_TOC_URL]	2011-03-03 17:50:0
180	180	[REDACTED]	[REDACTED] objectForKey:] [L:63] getting: k[Feed_TOC_URL] v{(null)}	2011-03-03 17:50:0
181	181	[REDACTED]	[REDACTED] [AppEnv initializeEnvironment] [L:240] gonna set KEY[Feed_TOC_URL] - [REDACTED]	2011-03-03 17:50:0

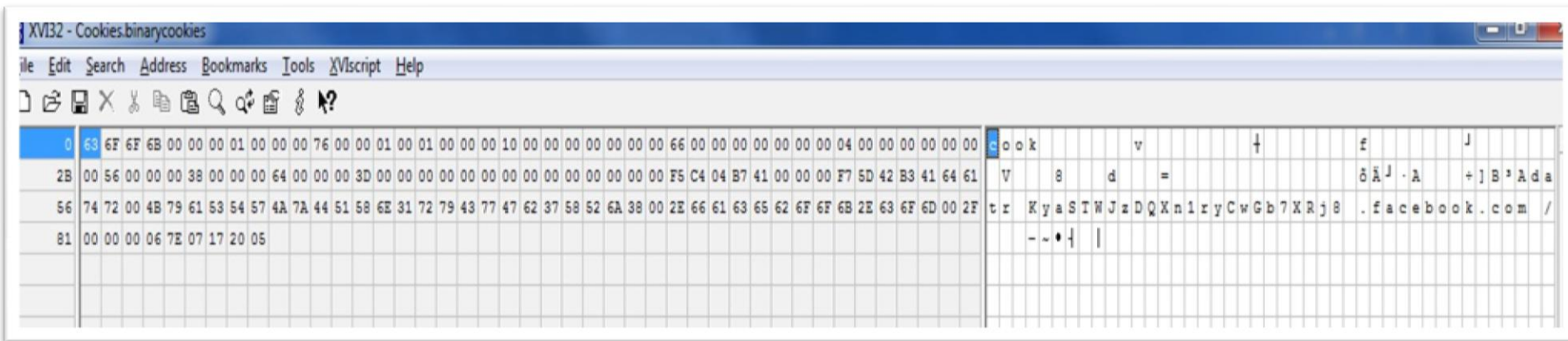
## 2. Insecure data storage

Example: iOS – Binary Cookies.



# 2. Insecure data storage

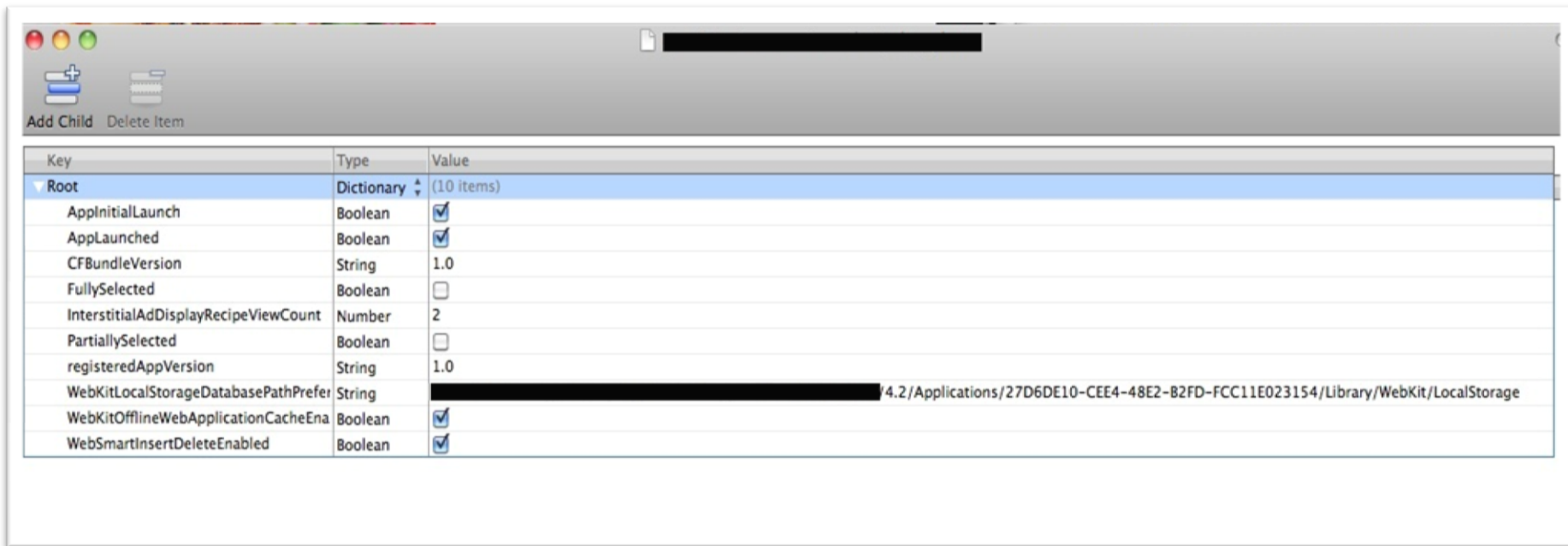
Example: iOS – Binary Cookies can be viewed using hex editors





## 2. Insecure data storage

Example: iOS – Property List Files can be viewed using any editor.



The screenshot shows a macOS-style window titled "Property List Editor" with a toolbar containing "Add Child" and "Delete Item" buttons. Below the toolbar is a table with three columns: "Key", "Type", and "Value". The table contains the following data:

Key	Type	Value
Root	Dictionary (10 items)	
AppInitialLaunch	Boolean	<input checked="" type="checkbox"/>
AppLaunched	Boolean	<input checked="" type="checkbox"/>
CFBundleVersion	String	1.0
FullySelected	Boolean	<input type="checkbox"/>
InterstitialAdDisplayRecipeViewCount	Number	2
PartiallySelected	Boolean	<input type="checkbox"/>
registeredAppVersion	String	1.0
WebKitLocalStorageDatabasePathPreference	String	~/Library/WebKit/LocalStorage
WebKitOfflineWebApplicationCacheEnabled	Boolean	<input checked="" type="checkbox"/>
WebSmartInsertDeleteEnabled	Boolean	<input checked="" type="checkbox"/>

## 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.

<http://www.wired.com/threatlevel/2010/07/intercepting-cell-phone-calls/>

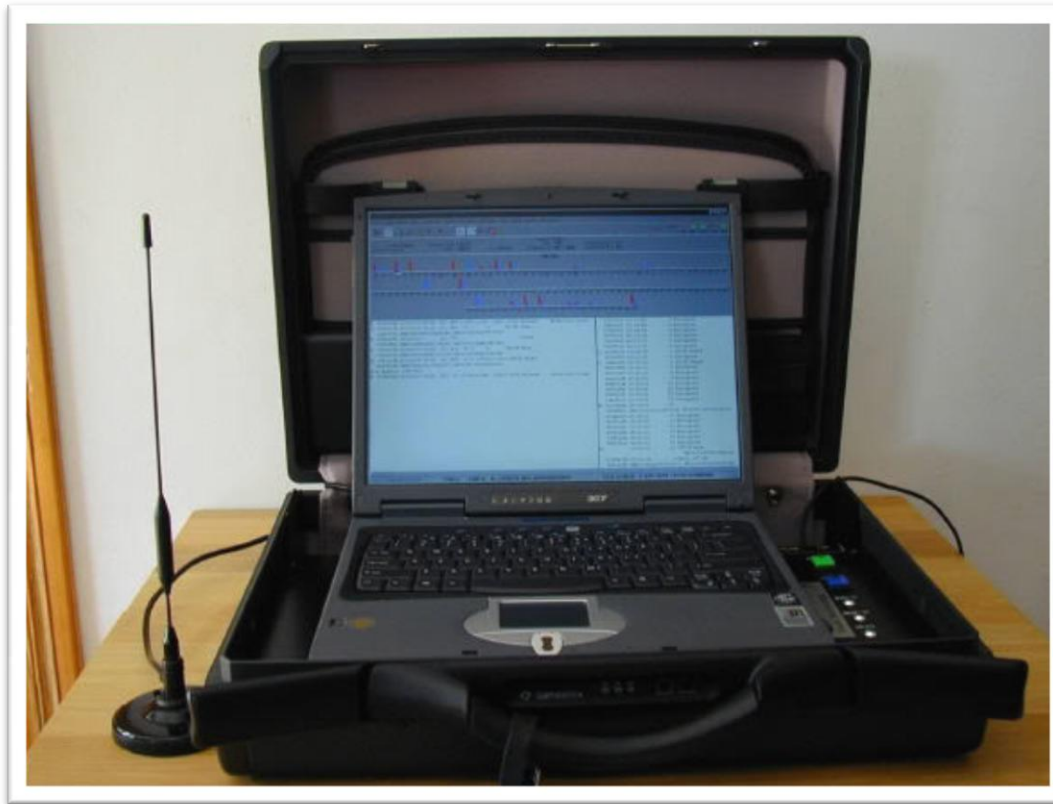
www.wired.com/threatlevel/2010/07/intercepting-cell-phone-calls/

PREVIOUS POST

**Hacker Spoofs Cell Phone Tower to Intercept Calls**

# 3. Insecure Communication

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



# 3. Insecure Communication

## Solution:

- For practical purposes lets 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.

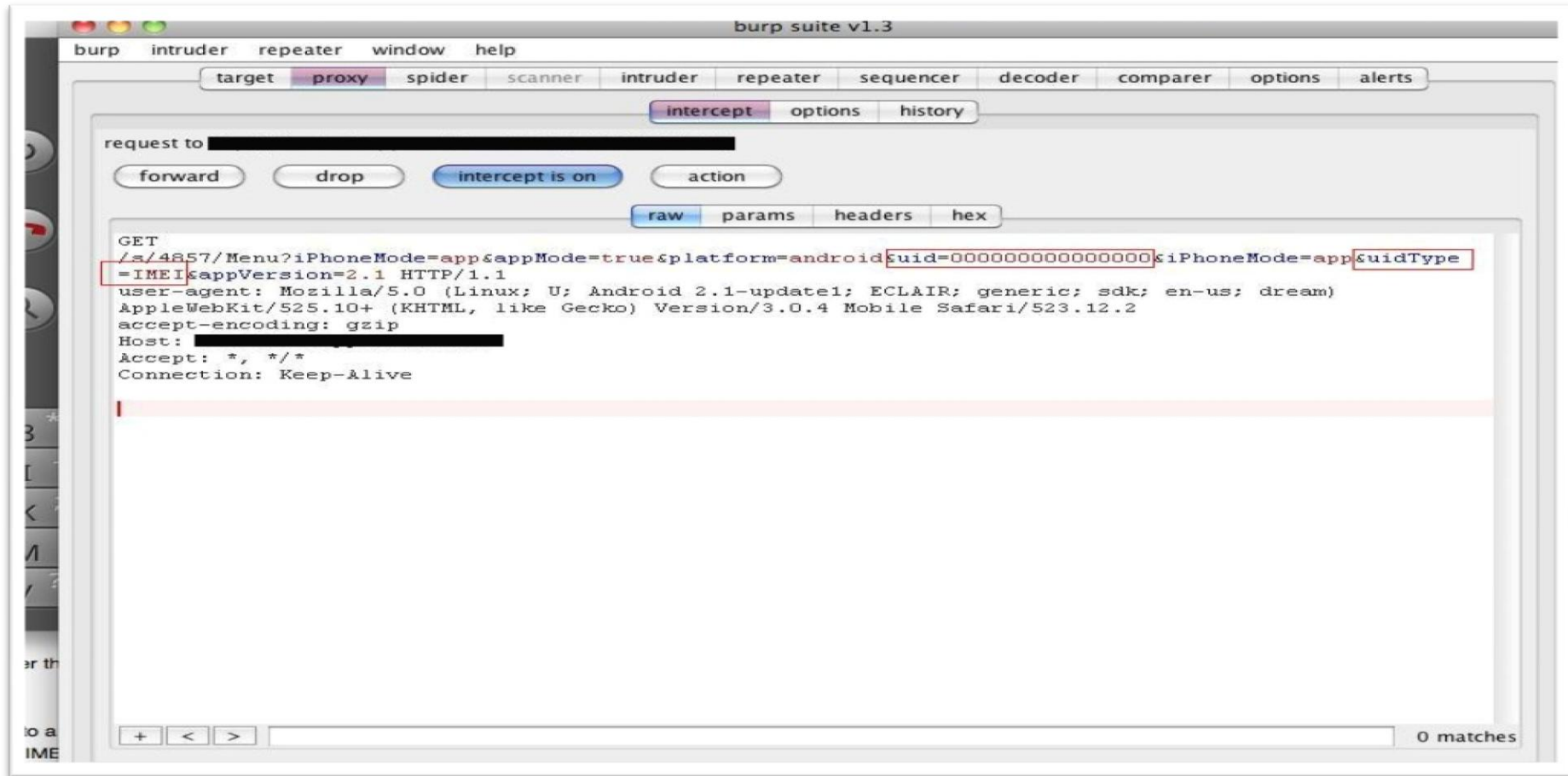
The screenshot shows the Burp Suite interface with a captured HTTP request to `http://m.twitter.com:80`. The request is a POST to `/oauth/authorize`. The body of the request contains several parameters, including `authenticity_token` and `session` parameters, which are highlighted in red. The `session` parameter contains the username and password in clear text: `session%5Busername_or_email%5D=` followed by a redacted username and `session%5Bpassword%5D=` followed by a redacted password. Other parameters include `oauth_token`, `ui=m`, and various cookies.

```
request to http://m.twitter.com:80 [128.121.146.100]
forward drop intercept is on action
raw params headers hex
POST /oauth/authorize HTTP/1.1
Host: m.twitter.com
Accept-Encoding: gzip
Accept-Language: en-US
Cookie: admobuu=429adbd5d8168afbf44fb124cbda0832; guest_id=128035786324618740; ui=m;
_twitter_sess=BAh7CjoMY3NyZl9pZCilOTNlMjE2MDI4YTg4NjZlNmVjMGU1ZWE2Y2NkYzJh%250AOTU6DnJldHVyb190byJga
HR0cDovL2OudHdpdHRlc15jb20vb2F1dGgvYXV0%250AaG9yaXp1P29hdXR0X3Rva2VuPUtSYk4xa0VSQ1dQS0ZQZ1pEQWdTW04
dWRk%250AdjB6bUt52212YXRKM0dKUT0PY3JlYXRlZFR9hdGwrCFCPRhsqAToHaWQiJTZh%250AZDUyMzM1YmUxNzYyNTIyOTQ1MD
Q1YjY1MzhkNmY4IgpmbGFzaE1DOidBY3Rp%250Ab25Db250cm9sbGVyOjppGhGFzaDo6Rmxhc2hiYXNoewAGOgpAdXNlZHsA--f98
ff6138866ee88fb42499159a20ab82d917e00; k=209.251.200.243.1280357861422844
Accept-Charset: utf-8, iso-8859-1, utf-16, *,q=0.7
Referer:
http://m.twitter.com/oauth/authorize?oauth_token=KRbN1kERCWPKFPgZDAGSem8uddv0zmKygmvatJ3GJQ
User-Agent: Mozilla/5.0 (Linux; U; Android 2.2; en-us; sdk Build/FRF91) AppleWebKit/533.1 (KHTML,
like Gecko) Version/4.0 Mobile Safari/533.1
Origin: http://m.twitter.com
Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Content-Type: application/x-www-form-urlencoded
Content-Length: 201

authenticity_token=7dced1ca0e85bc94765ee4c46a79a70cca609235&session%5Busername_or_email%5D=
[REDACTED]&session%5Bpassword%5D=[REDACTED]&oauth_token=KRbN1kERCWPKFPgZDAGSem8uddv0zmKygmvatJ3GJ
Q
```

# 3. Insecure Communication

Example: Android App sending IMEI number in clear text.





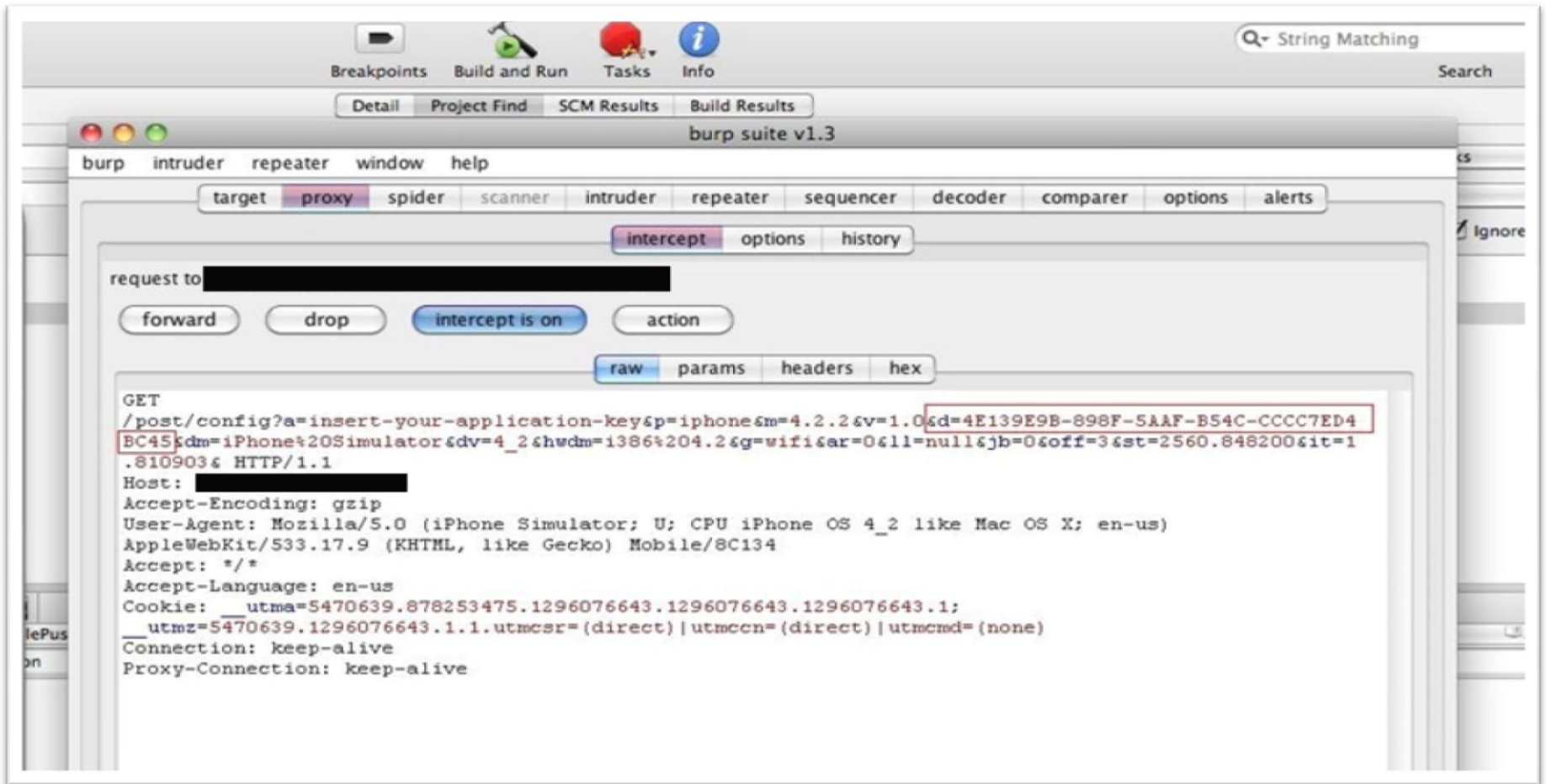
# 3. Insecure Communication

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

```
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: [REDACTED]
Request Version: HTTP/1.1
Host: [REDACTED]
User-Agent: [REDACTED]
Accept: */*\r\n
Accept-Language: en-us\r\n
Accept-Encoding: gzip, deflate\r\n
Cookie: PFUID=400c37834d6fd4d254011000ffffff9d\r\n
▶ Authorization: Basic [REDACTED]
Connection: keep-alive\r\n
```

# 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

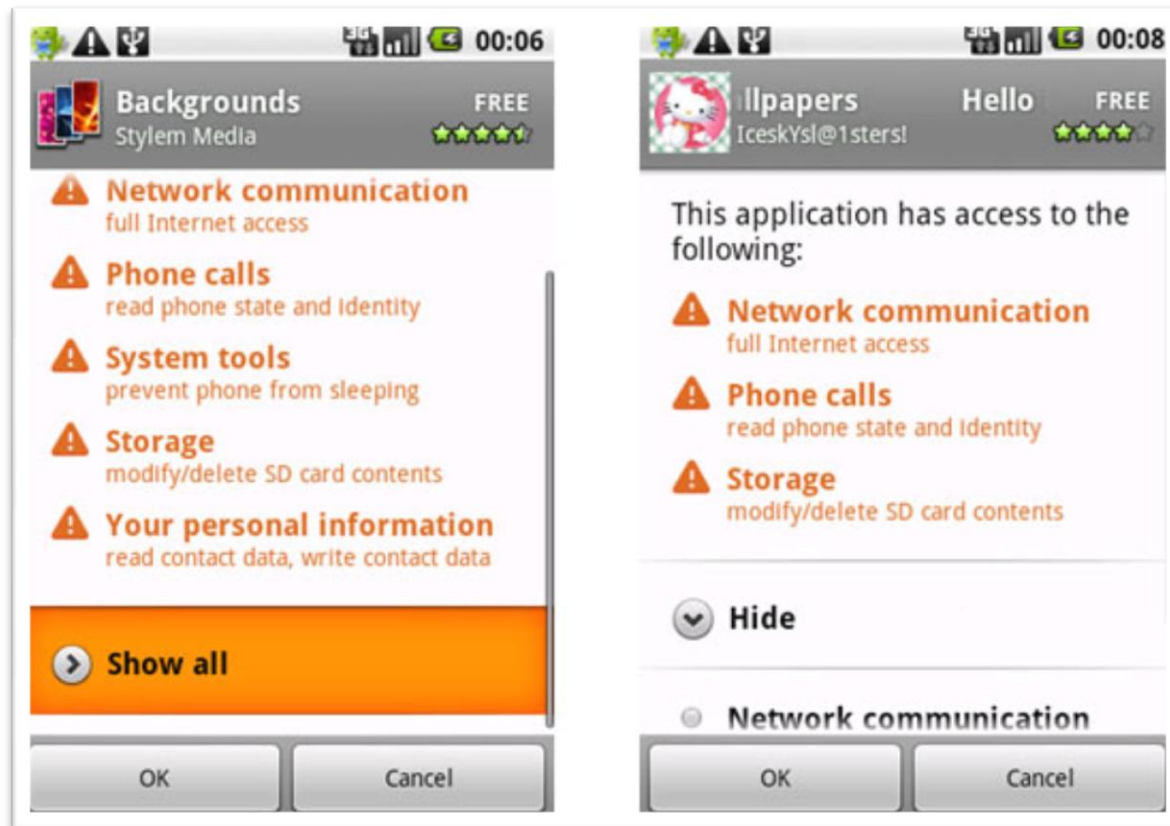
- ▶ Transparent - all the permissions an app has access to
- ▶ Permissions can be found during Installation
- ▶ AndroidManifest.XML file contains the permission details
- ▶ Can be viewed anytime under Managing Applications

### ■ iOS

- ▶ Not so transparent
- ▶ If the app has access to GPS there is a 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



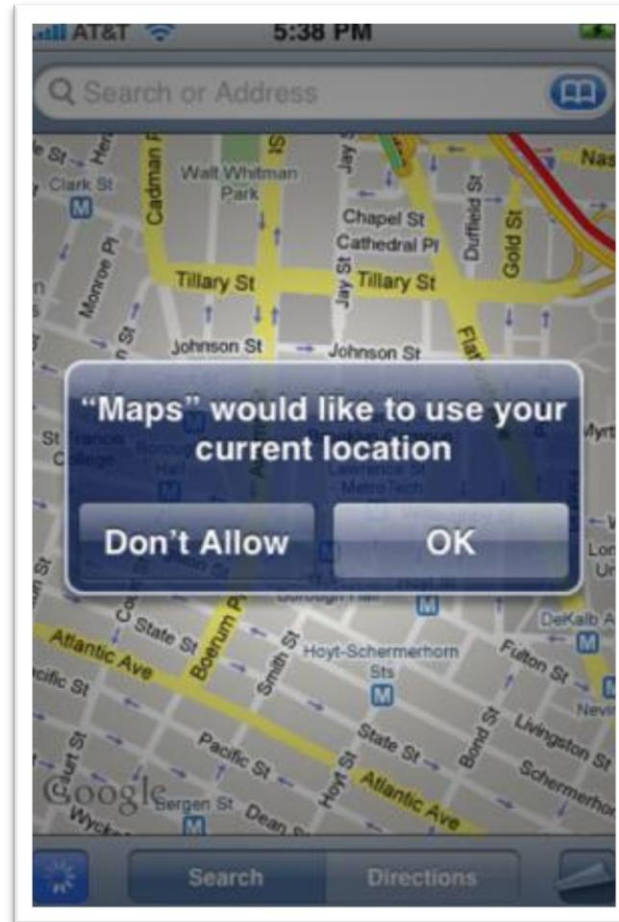
# 4. Excessive Permissions

Example: Android – Sample AndroidManifest.XML file.

```
    </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.

```
request response
raw params headers hex
POST /users/savelocation HTTP/1.1
Host: ██████████
User-Agent: ██████████ (iPad1,1; iPhone OS 4.3.3; en_US)
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept-Encoding: gzip
Content-Length: 90
Cookie: utma=173415231.1866741094.1309238702.1309238702.1309238702.1; __utmz=173415231.1309238702.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
██████████ cpw24-025bxas3y4_i=2; cpw24-025bxas3y4_vt=1309238771307; cpw24-025bxas3y4_r=0; cpw24-025bxas3y4_s=188669186
Connection: keep-alive
Proxy-Connection: keep-alive

device_id=██████████9664be737e3&latitude=██0.863849&longitude=-██3.959271
```

# 4. Excessive Permissions

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

```
HTTP/1.1 200 OK
Date: Tue, 28 Jun 2011 13:49:03 GMT
Server: Apache/2.2.14 (Ubuntu)
X-Powered-By: PHP/5.3.2-1ubuntu4.7
Vary: Accept-Encoding
Content-Length: 2648
Connection: close
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0"><dict><key>comments</key><array><dict><key>id</key><string>11154436</string><key>type_id</key><string>7</string><key>body</key><string>Sky is blue lat
</string><key>device_id</key><string>7809866</string><key>created</key><string>2011-06-27
21:48:42</string><key>parent_id</key><string>0</string><key>item_id</key><string>1709</string><key>app_id</key><string>1709</string><key>imgpath</key><string></string><key>1
at</key><string>0.863987</string><key>lon</key><string>-3.959335</string><key>profile_image</key><string>http://img.tweetimag.es/i/aj<img alt="redacted profile image" data-bbox="815 485 845 505"/></string><key>nickname</key><
string>aj<img alt="redacted nickname" data-bbox="815 505 845 525"/></string><key>guid</key><string>11154436</string><key>comment_type</key><string>1709</string><key>replies</key><string>1</string><key>timeago</key><string>9
hours
ago</string><key>user_id</key><string>7809866</string></dict><dict><key>id</key><string>11113236</string><key>type_id</key><string>7</string><key>body</key><string>this is
it
</string><key>device_id</key><string>7837127</string><key>created</key><string>2011-06-27
07:45:49</string><key>parent_id</key><string>0</string><key>item_id</key><string>1709</string><key>app_id</key><string>1709</string><key>imgpath</key><string></string><key>1
at</key><string>0.000000</string><key>lon</key><string>0.000000</string><key>profile_image</key><string>http://img.tweetimag.es/i/C<img alt="redacted profile image" data-bbox="815 585 845 605"/></string><key>nickname</key><
string>C<img alt="redacted nickname" data-bbox="815 605 845 625"/></string><key>guid</key><string>11113236</string><key>comment_type</key><string>1709</string><key>replies</key><string>1</string><key>timeago</key><string>
>23 hours
ago</string><key>user_id</key><string>7837127</string></dict><dict><key>id</key><string>10421209</string><key>type_id</key><string>7</string><key>body</key><string>test
</string><key>device_id</key><string>2702319</string><key>created</key><string>2011-06-14
17:41:38</string><key>parent_id</key><string>0</string><key>item_id</key><string>1709</string><key>app_id</key><string>1709</string><key>imgpath</key><string></string><key>1
at</key><string>4.066360</string><key>lon</key><string>-18.380920</string><key>profile_image</key><string>http://img.tweetimag.es/i/m<img alt="redacted profile image" data-bbox="815 685 845 705"/></string><key>nickname</key><k
ey><string>m<img alt="redacted nickname" data-bbox="815 705 845 725"/></string><key>guid</key><string>10421209</string><key>comment_type</key><string>1709</string><key>replies</key><string>3</string><key>timeago</key><st
ring>2 weeks ago</string><key>user_id</key><string>2702319</string></dict></array><key>total</key><string>3</string></dict></plist>
```

# 4. Excessive Permissions

Examples: iOS App constantly recording GPS location of the user



# 4. Excessive Permissions

The screenshot displays the Xcode interface. On the left, the 'Link Binary With Libraries' list is visible, with a red box highlighting the entire list and a blue box highlighting 'CoreLocation.framework'. The right pane shows source code with several CoreLocation imports. A blue box highlights the 'MedialetsAnalyticsManager.h' header file. A red box highlights the 'CoreLocation.framework' entry in the library list. A black box redacts a portion of the code.

```
#import <CoreLocation/CLLocationManager.h>
#import <CoreLocation/CLLocationManagerDelegate.h>
#endif /* __CORELOCATION__ */

MedialetsAnalyticsManager.h
#import <CoreLocation/CoreLocation.h>
#import <CoreLocation/CoreLocation.h>
[REDACTED]
#import <CoreLocation/CoreLocation.h>
#import <CoreLocation/CoreLocation.h>

CLLocationHeading.h
* CoreLocation
#import <CoreLocation/CLLocation.h>

CLLocationErrorDomain.h
* CoreLocation
* Error returned as the domain to NSError from CoreLocation.

CLLocationManagerDelegate.h
* CoreLocation
#import <CoreLocation/CLLocationManager.h>
```

# 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](mailto:praveen.nallasamy@owasp.org)  
[praveen.nallasamy@gmail.com](mailto:praveen.nallasamy@gmail.com)

