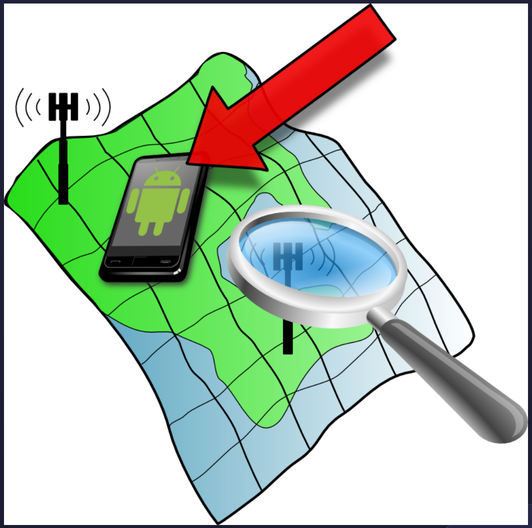


Android geolocation using GSM network

« Where was Waldroid? »

Renaud Lifchitz
renaud.lifchitz+27c3@gmail.com

#27c3
27-30 December 2010, Berlin

Speaker's bio

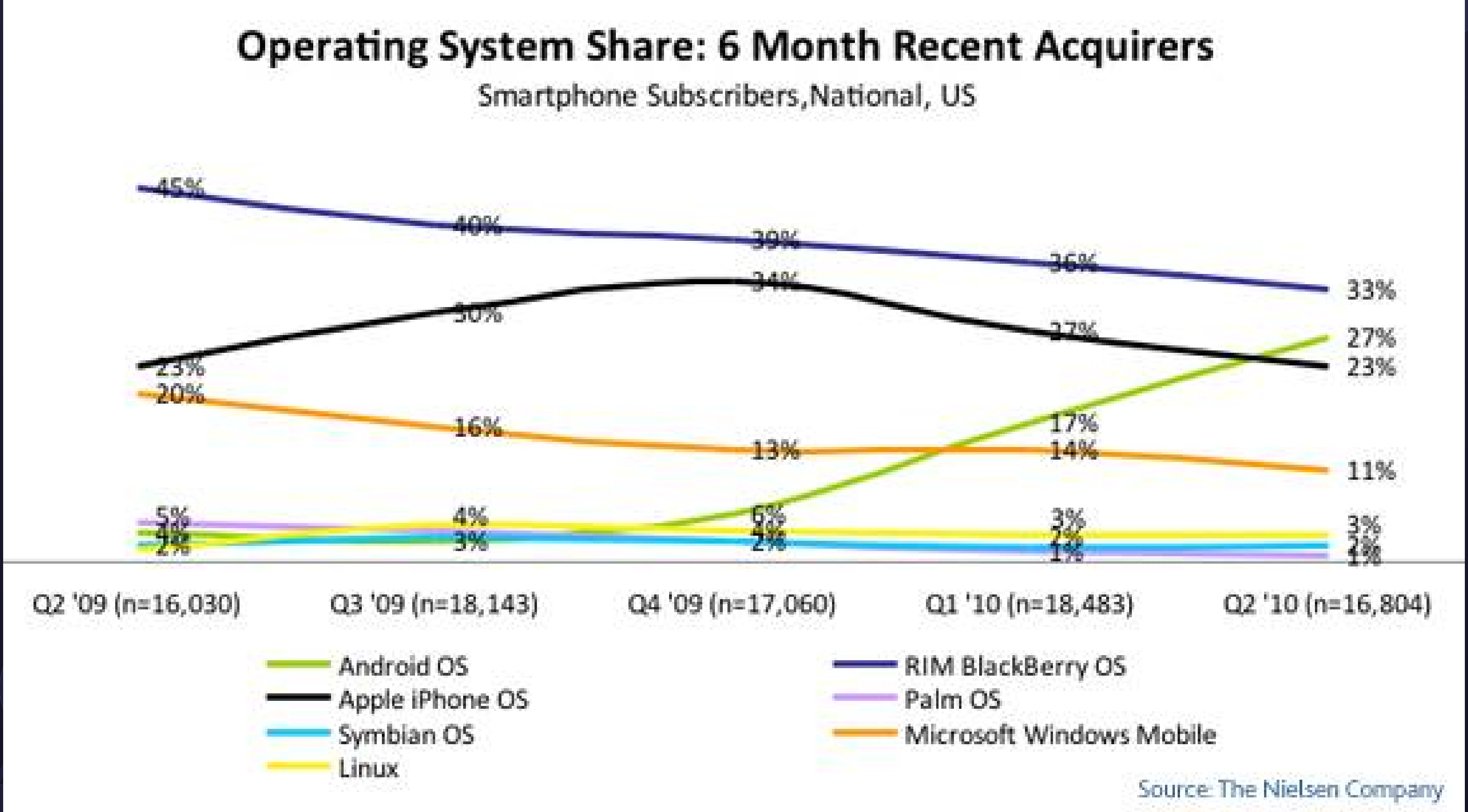
- French computer security engineer
- Main activities:
  - Penetration testing&security audits
  - Security trainings
  - Security research
- Main interests:
  - Security of protocols (authentication, cryptography, information leakage, zero-knowledge proofs...)
  - Number theory (integer factorization, primality tests, elliptic curves)

Why Android?

Why Android?

- Why not?
- In just 2 years, 300,000 Android phones activated each day
  (Andy Rubin, Google, 2010/12/09)
- Android sales overtake iPhone in the U.S. since summer
- Because hacking on Android is sooooo cool (Linux kernel ☺)

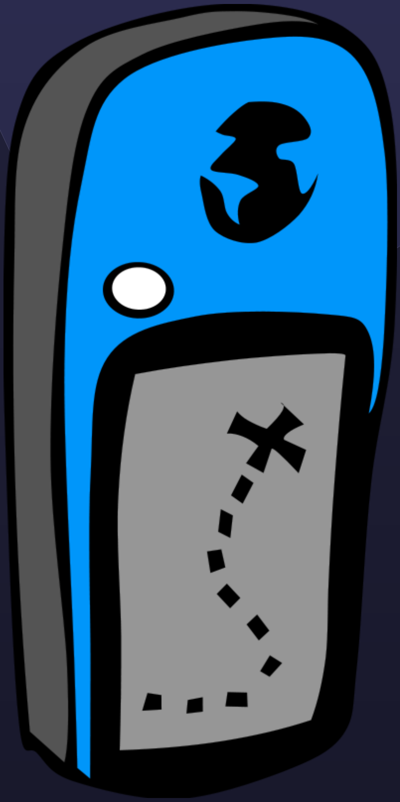
Why Android?



Source: The Nielsen Company

Geolocation: different approaches

GPS



- Pros:
  - Very accurate
- Cons:
  - Phone needs a built-in GPS
  - User must switch it on
  - Doesn't work inside buildings nor underground

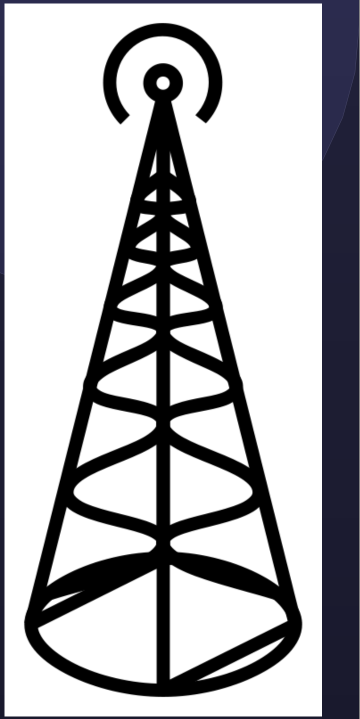
Wi-Fi

- Pros:
  - Works inside buildings
- Cons:
  - Phone needs built-in Wi-Fi
  - User must switch it on
  - Less accurate than GPS
  - Needs access points

GSM location

- Pros:
  - No need for built-in GPS or Wi-Fi
  - Can be done from the network side
- Cons:
  - Medium accuracy
  - Needs GSM coverage

Cell location resolution

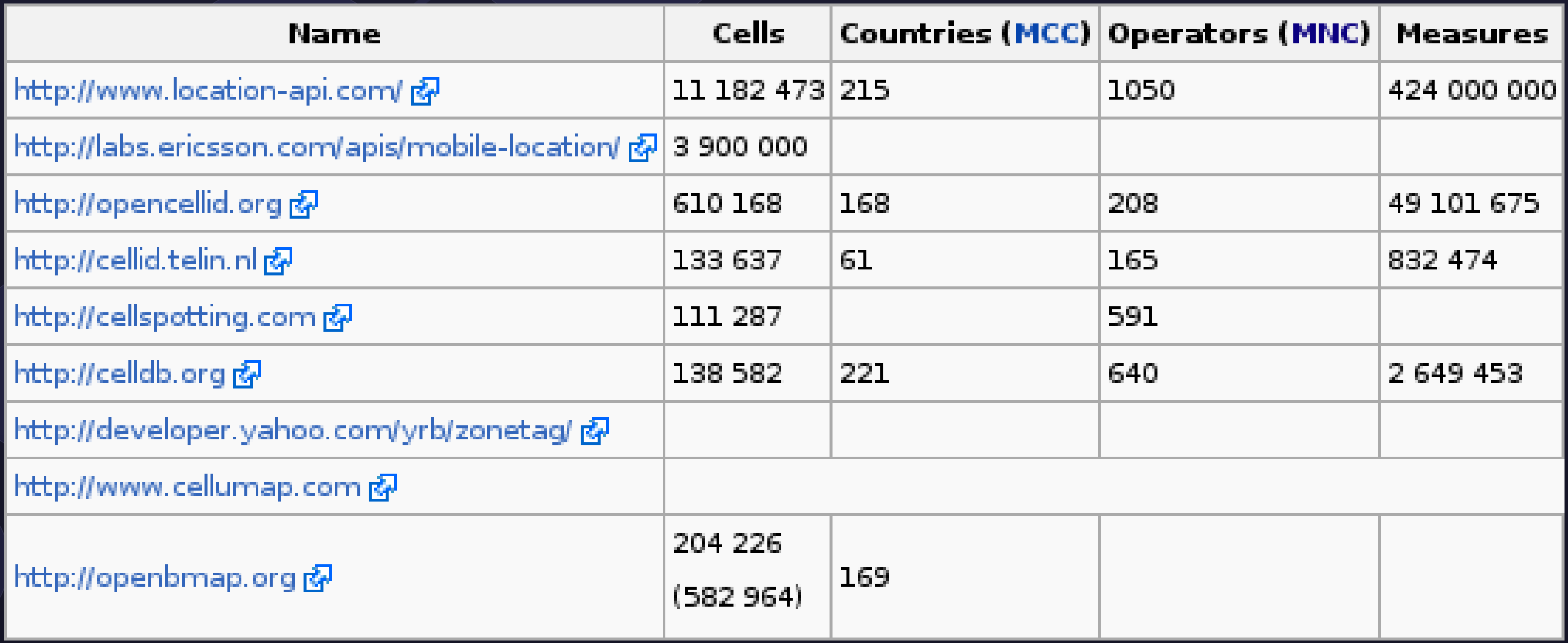


- Every GSM cell (BTS) is identified by 4 numbers:
  - MCC: Mobile Country Code
  - MNC: Mobile Network Code
  - LAC: Location Area Code
  - CID: Cell ID

☞ (MCC: 262, MNC: 01) = T-Mobile® Deutschland

Cell location resolution

- There have been several attempts to build databases of GSM cells:



Name	Cells	Countries (MCC)	Operators (MNC)	Measures
http://www.location-api.com/	11 182 473	215	1050	424 000 000
http://labs.ericsson.com/apis/mobile-location/	3 900 000			
http://opencellid.org	610 168	168	208	49 101 675
http://cellid.telin.nl	133 637	61	165	832 474
http://cellspotting.com	111 287		591	
http://celldb.org	138 582	221	640	2 649 453
http://developer.yahoo.com/yrb/zonetag/				
http://www.cellumap.com				
http://openbmap.org	204 226 (582 964)	169		

Source: Wikipedia (http://en.wikipedia.org/wiki/Cell_ID)

Cell location resolution

- Why not use Google fantastic indexing power?
- Huge and continuously updated database thanks to:

Flickr photo by PhyreWorX
Licensed under theCC Attribution-Share Alike 2.0 Generic license

Google cars

&

Android phones

Cell location resolution

- Google API? Quite confidential...
- Reverse-engineer:
  - What is used when you run Android Google Maps without GPS nor Wi-Fi
  - What is used by Google Gears plugin when you do a Google local search in your browser

Cell location resolution

- Android Google Maps internals:
  - tcpdump ARM compilation
  - Proprietary binary protocol
  - HTTP POSTed to http://www.google.com/glm/mmap
  - See “Poor Man's GPS” by Dhaval Motghare for reference: http://www.orangeapple.org/?p=82
  - Buggy...

Cell location resolution

- Google Gears internals:
  - Sniff Firefox plugin network traffic
  - See it's simple JSON!
  - Some (confidential!) reference here: http://code.google.com/p/gears/wiki/GeolocationAPI
  - “Officially deprecated” but updated and works a lot better than previous binary protocol

Cell location resolution

```
 POST /loc/json HTTP/1.1
 Accept-Charset: utf-8
 Accept-Encoding: plain
 Cache-Control: no-cache
 Connection: close
 Content-Length: 242
 Content-Type: application/json
 Host: www.google.com

 {"radio_type": "gsm", "address_language": "fr_FR",
"host": "maps.google.com", "version": "1.1.0",
"cell_towers": [{"mobile_network_code": 1, "cell_id":
32755, "mobile_country_code": 208, "location_area_code":
24832}], "request_address": true}
```

Google Gears GSM Geolocation API full query

Cell location resolution

```
 {"location":
{"latitude":48.886363,"longitude":2.246213,"address":
{"country":"France","country_code":"FR","region":"Ile-de-
France","county":"Hauts-de-
Seine","city":"Puteaux","street":"Rue Paul
Lafargue","street_number":"16","postal_code":"92800"},"acc
uracy":500.0},"access_token":"2:1dxrwvFk6ejLzSpv:BDHb9oizx
wm0bwsb"}
```

Google Gears GSM Geolocation API response body

- Interesting details:
  - Latitude&longitude
  - Full human-readable address (including street number, street name, zip code, city, region and country!)
  - Accuracy (in meters) → cell coverage?

Cell location resolution

- Going further: mapping the GSM network using sniffing with a SDR (Software Defined Radio) or an old phone (Nokia 3310)
- USRP 1 from Ettus Research LLC:

Cell location resolution

- Use excellent AirProbe project:

  https://svn.berlin.ccc.de/projects/airprobe/

1. Scan with GnuRadio
2. Demodulate with AirProbe
3. Decode with Wireshark

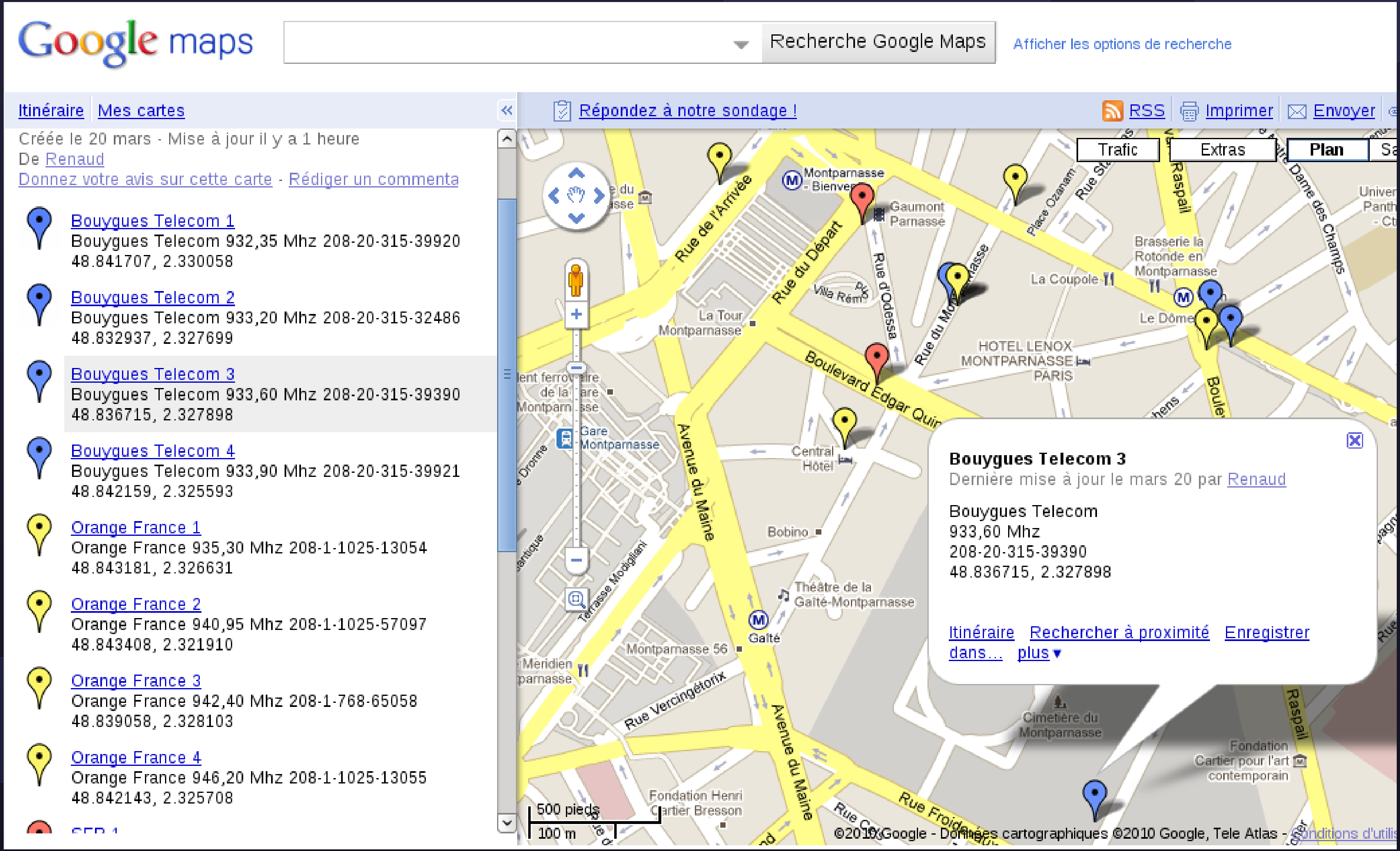
Cell location resolution

```
$ tshark -V gsm_a.cell_ci -r out1.xml | grep -A2 'Cell CI'
    Cell CI: 0x3198 (12696)
    Location Area Identification - LAC (0x1005)
        Mobile Country Code (MCC): 208, Mobile Network Code (MNC): 10
--
    Cell CI: 0x31fe (12798)
    Location Area Identification - LAC (0x1005)
        Mobile Country Code (MCC): 208, Mobile Network Code (MNC): 10
--
    Cell CI: 0x3806 (14342)
    Location Area Identification - LAC (0x044c)
        Mobile Country Code (MCC): 208, Mobile Network Code (MNC): 10
--
    Cell CI: 0xe0ba (57530)
    Location Area Identification - LAC (0x044c)
        Mobile Country Code (MCC): 208, Mobile Network Code (MNC): 10
```

Cell ID extraction from a demodulated capture

Cell location resolution

- Result!:



Créée le 20 mars - Mise à jour il y a 1 heure
De Renaud
Donnez votre avis sur cette carte - Rédiger un commenta

Bouygues Telecom 1
Bouygues Telecom 932,35 Mhz 208-20-315-39920
48.841707, 2.330058

Bouygues Telecom 2
Bouygues Telecom 933,20 Mhz 208-20-315-32486
48.832937, 2.327699

Bouygues Telecom 3
Bouygues Telecom 933,60 Mhz 208-20-315-39390
48.836715, 2.327898

Bouygues Telecom 4
Bouygues Telecom 933,90 Mhz 208-20-315-39921
48.842159, 2.325593

Orange France 1
Orange France 935,30 Mhz 208-1-1025-13054
48.843181, 2.326631

Orange France 2
Orange France 940,95 Mhz 208-1-1025-57097
48.843408, 2.321910

Orange France 3
Orange France 942,40 Mhz 208-1-768-65058
48.839058, 2.328103

Orange France 4
Orange France 946,20 Mhz 208-1-1025-13055
48.842143, 2.325708

SFR 1

Bouygues Telecom 3
Dernière mise à jour le mars 20 par Renaud

Bouygues Telecom
933,60 Mhz
208-20-315-39390
48.836715, 2.327898

Itinéraire Rechercher à proximité Enregistrer dans... plus ▾

GSM mapping 1 square kilometre of Paris from my bed ☺

Attack vectors

Attack basics

- Android uses a specific logging facility
- Enabled by default
- 3 or 4 different logs
- Circular memory buffers
- Handled by character device files
- Built-in `logcat` tool to manipulate the logs

Attack basics

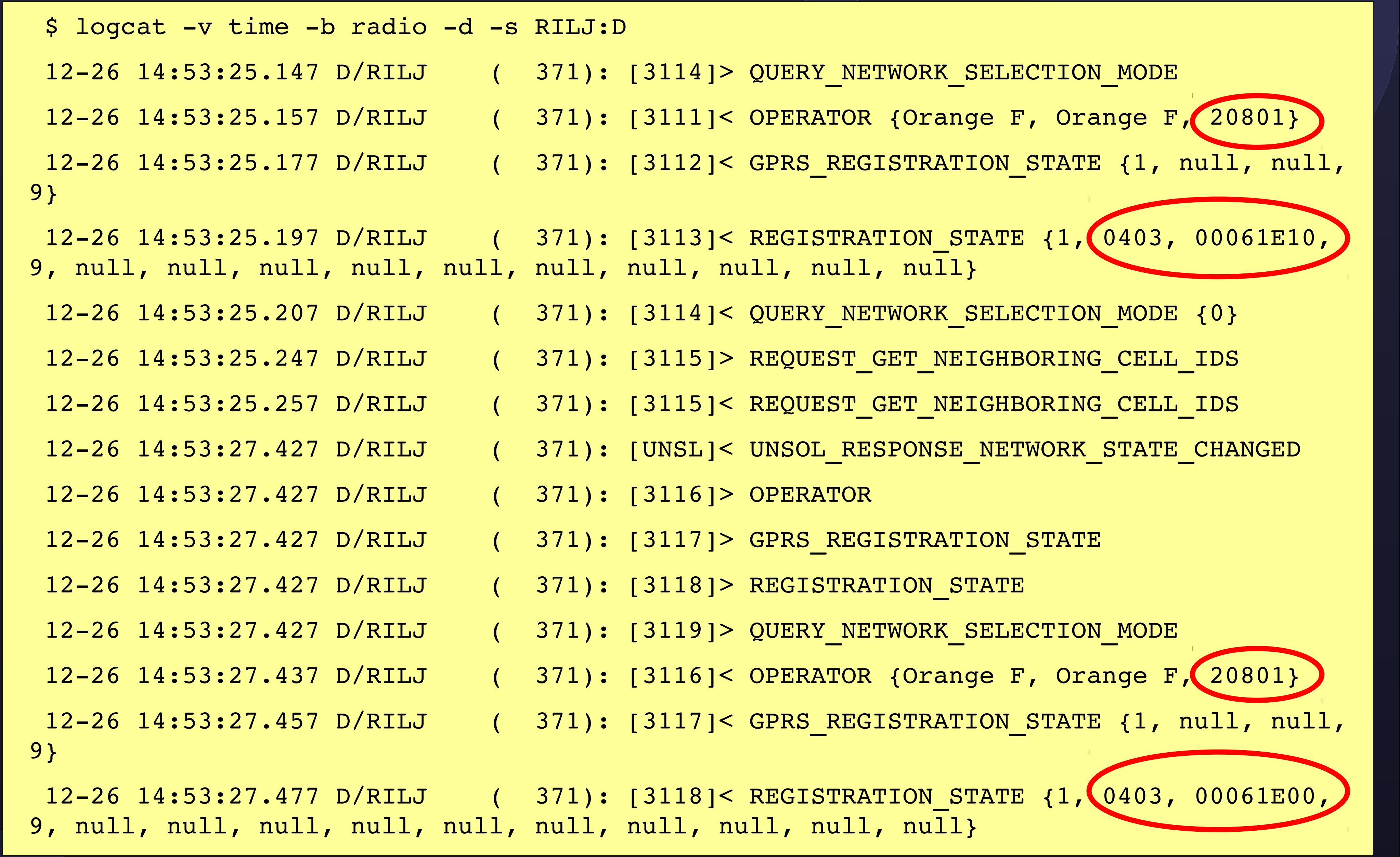
```
 # ls -l /dev/log
 crw-rw--w-    1 root     log        10,  36 Dec 25 15:15 system
 crw-rw--w-    1 root     log        10,  37 Dec 25 15:15 radio
 crw-rw--w-    1 root     log        10,  39 Dec 25 15:15 main
 crw-rw--w-    1 root     log        10,  38 Dec 25 15:15 events

 # cd /dev/log ; for f in *; do logcat -b $f -g; done
 /dev/log/events: ring buffer is 256Kb (255Kb consumed), max entry is 4096b, max
payload is 4076b
 /dev/log/main: ring buffer is 64Kb (63Kb consumed), max entry is 4096b, max
payload is 4076b
 /dev/log/radio: ring buffer is 64Kb (14Kb consumed), max entry is 4096b, max
payload is 4076b
 /dev/log/system: ring buffer is 64Kb (6Kb consumed), max entry is 4096b, max
payload is 4076b
```

Playing with logging facility

Attack basics

```
# hexdump -C radio | head
00000000  4e 00 00 00 73 01 00 00  95 01 00 00 8c 3f 17 4d  |N...s........?.M|
00000010  81 31 51 12 03 47 53 4d  00 5b 47 73 6d 44 61 74  |.1Q..GSM.[GsmDat|
00000020  61 43 6f 6e 6e 65 63 74  69 6f 6e 2d 31 5d 20 44  |aConnection-1] D|
00000030  63 49 6e 61 63 74 69 76  65 53 74 61 74 65 3a 20  |cInactiveState: |
00000040  73 65 74 45 6e 74 65 72  4e 6f 74 69 63 61 74 69  |setEnterNoticati|
00000050  6f 6e 50 61 72 61 6d 73  20 63 70 2c 63 61 75 73  |onParams cp,caus|
00000060  65 00 47 00 fa d3 73 01  00 00 95 01 00 00 8c 3f  |e.G...s........?|
00000070  17 4d 81 31 51 12 03 47  53 4d 00 5b 47 73 6d 44  |.M.1Q..GSM.[GsmD|
00000080  61 74 61 43 6f 6e 6e 65  63 74 69 6f 6e 2d 31 5d  |ataConnection-1]|
00000090  20 44 63 41 63 74 69 76  65 53 74 61 74 65 3a 20  | DcActiveState: |
```

Playing with logging facility



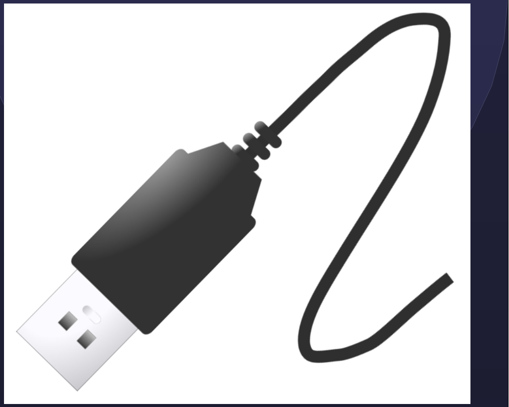
```
 $ logcat -v time -b radio -d -s RILJ:D
 12-26 14:53:25.147 D/RILJ    (  371): [3114]> QUERY_NETWORK_SELECTION_MODE
 12-26 14:53:25.157 D/RILJ    (  371): [3111]< OPERATOR {Orange F, Orange F, 20801}
 12-26 14:53:25.177 D/RILJ    (  371): [3112]< GPRS_REGISTRATION_STATE {1, null, null,
9}
 12-26 14:53:25.197 D/RILJ    (  371): [3113]< REGISTRATION_STATE {1, 0403, 00061E10,
9, null, null, null, null, null, null, null, null, null, null}
 12-26 14:53:25.207 D/RILJ    (  371): [3114]< QUERY_NETWORK_SELECTION_MODE {0}
 12-26 14:53:25.247 D/RILJ    (  371): [3115]> REQUEST_GET_NEIGHBORING_CELL_IDS
 12-26 14:53:25.257 D/RILJ    (  371): [3115]< REQUEST_GET_NEIGHBORING_CELL_IDS
 12-26 14:53:27.427 D/RILJ    (  371): [UNSL]< UNSOL_RESPONSE_NETWORK_STATE_CHANGED
 12-26 14:53:27.427 D/RILJ    (  371): [3116]> OPERATOR
 12-26 14:53:27.427 D/RILJ    (  371): [3117]> GPRS_REGISTRATION_STATE
 12-26 14:53:27.427 D/RILJ    (  371): [3118]> REGISTRATION_STATE
 12-26 14:53:27.427 D/RILJ    (  371): [3119]> QUERY_NETWORK_SELECTION_MODE
 12-26 14:53:27.437 D/RILJ    (  371): [3116]< OPERATOR {Orange F, Orange F, 20801}
 12-26 14:53:27.457 D/RILJ    (  371): [3117]< GPRS_REGISTRATION_STATE {1, null, null,
9}
 12-26 14:53:27.477 D/RILJ    (  371): [3118]< REGISTRATION_STATE {1, 0403, 00061E00,
9, null, null, null, null, null, null, null, null, null, null}
```

History of user's visited MCCs+MNCs, LACs, CIDs in `radio` logs

Attack basics

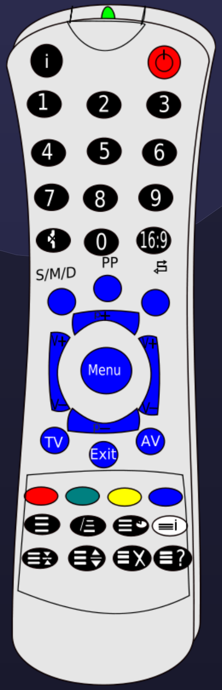
- Attack scenario:
  - Collect history of visited GSM cells on the victim's side (no prior access needed)
  - Send them to the attacker
  - Resolve them into latitude&longitude
- Attack range:
  - Local (i.e. physical attack)
  - Remote (here remote means using a local vulnerability!)

Physical attack



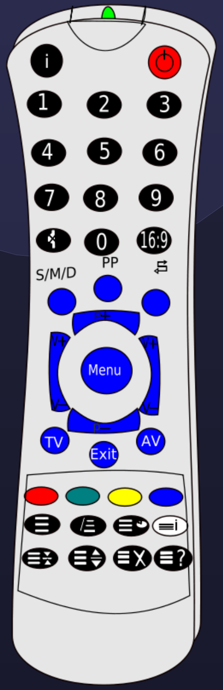
- Connect the victim's phone to the attacker computer via USB
- Requires:
  - Physical access to the victim's phone for a few seconds
- Works even if the victim's phone is locked! (using USB debugging function)

Remote attack



- Remotely spy the victim
- Malware application who abuse either:
  - User trust
  - Android security model
- Requires:
  - A bit of social engineering (or not ☺)

Remote attack



- Android permissions model: Dalvik (java) sandbox
- Permissions: android.permission.*
- What can a user fear?
  - Dangerous combination of 2 permissions:

    ACCESS_COARSE_LOCATION
    or ACCESS_FINE_LOCATION

    + INTERNET

Remote attack

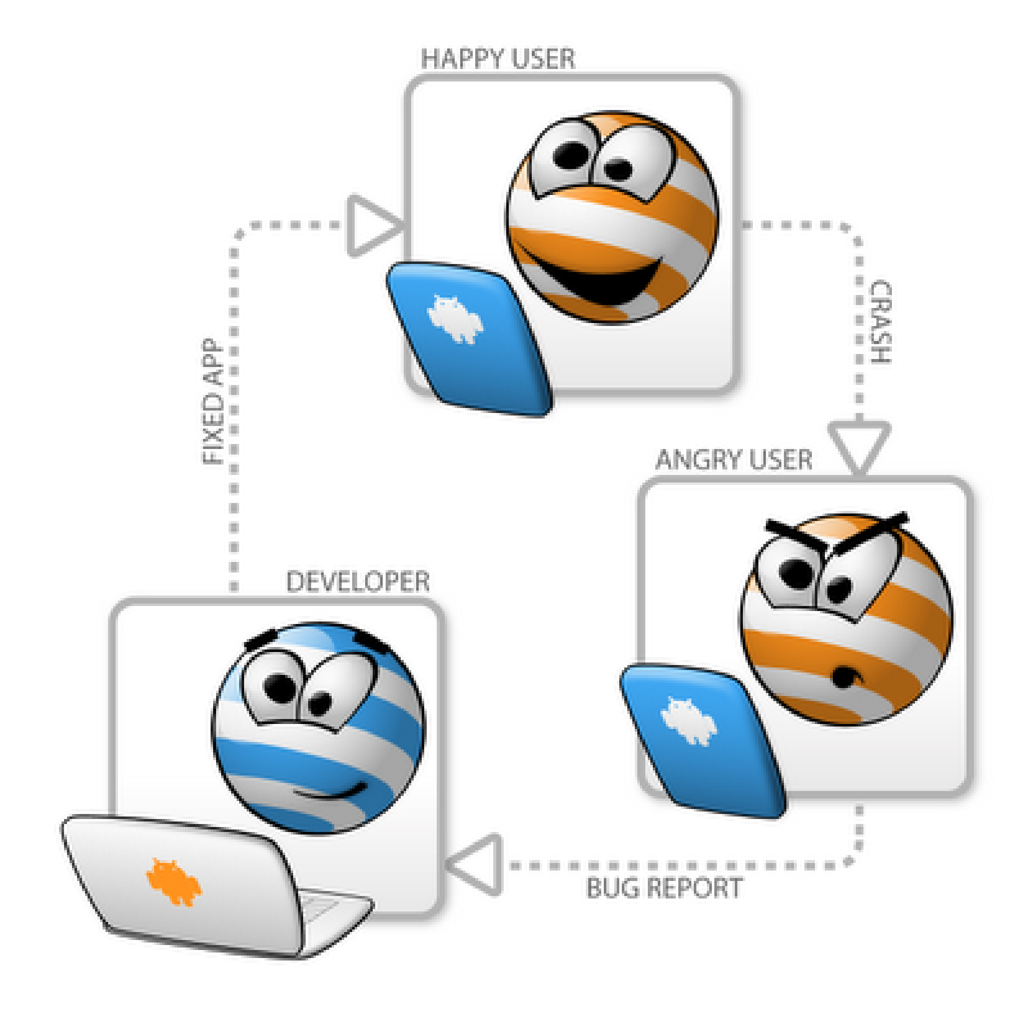
- 1st attack - Use both permissions:
  - Internet permission is needed for free ad-sponsored applications
  - Official geolocation permission is needed for location-aware applications

☞ most users won't care!

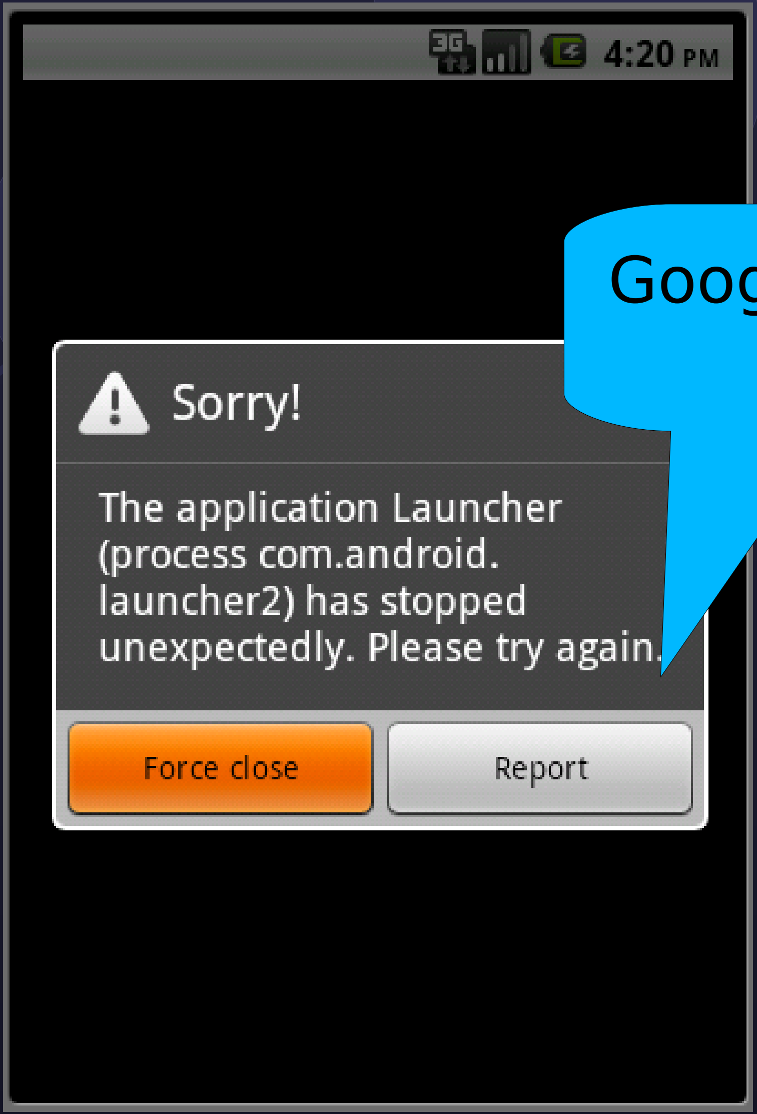
Remote attack

- 2nd attack – Use the radio logs:
  - Instead of using Android geolocation API, read radio logs (READ_LOGS permission) to collect Cell Ids
  - Write results into the system log (no permission needed!)
  - Voluntarily crash the application when needed (no permission needed!)
  - If the user reports the crash, system log is sent to the developer using the integrated Google Feedback client ☺

Remote attack



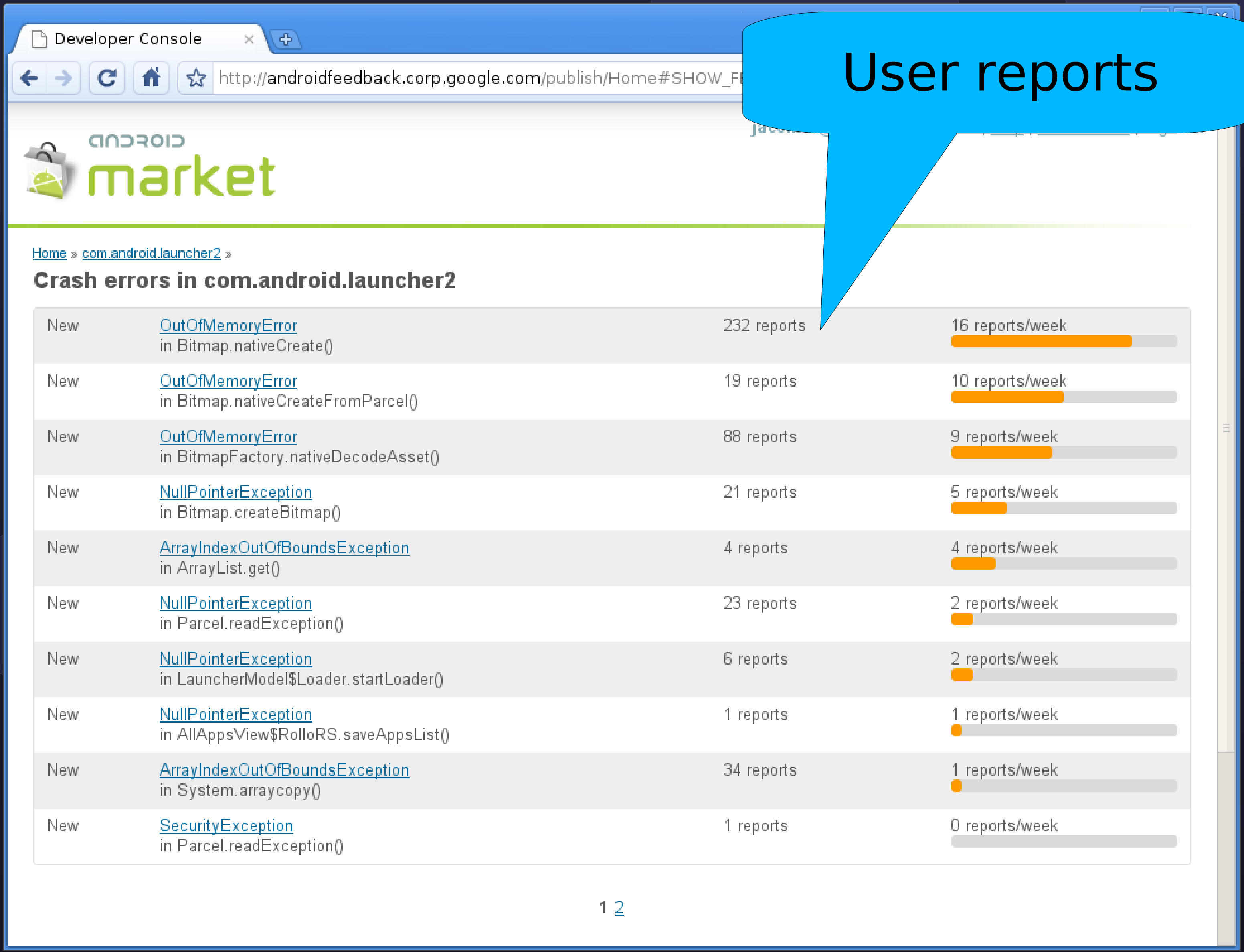
Remote attack



The application Launcher (process com.android. launcher2) has stopped unexpectedly. Please try again.

Google Feedback client

Remote attack



User reports

Home » com.android.launcher2 »

Crash errors in com.android.launcher2

New	OutOfMemoryError in Bitmap.nativeCreate()	232 reports	16 reports/week
New	OutOfMemoryError in Bitmap.nativeCreateFromParcel()	19 reports	10 reports/week
New	OutOfMemoryError in BitmapFactory.nativeDecodeAsset()	88 reports	9 reports/week
New	NullPointerException in Bitmap.createBitmap()	21 reports	5 reports/week
New	ArrayIndexOutOfBoundsException in ArrayList.get()	4 reports	4 reports/week
New	NullPointerException in Parcel.readException()	23 reports	2 reports/week
New	NullPointerException in LauncherModel$Loader.startLoader()	6 reports	2 reports/week
New	NullPointerException in AllAppsView$RolloRS.saveAppsList()	1 reports	1 reports/week
New	ArrayIndexOutOfBoundsException in System.arraycopy()	34 reports	1 reports/week
New	SecurityException in Parcel.readException()	1 reports	0 reports/week

1 2

Remote attack

- 3rd attack - Use Android NDK to completely bypass permissions model:
  - Native Development Kit allows developer to call native functions (C/C++ code) from their applications (similar to JNI)
  - Works outside the Dalvik sandbox...
- Arbitrary file access, code execution, network access... ☺

Remote attack

Last minute idea!

- 4th attack – Man-in-The-Middle attack during application download over Wi-Fi:
  - The new Android Market&Android Download Manager send application name, description, permissions then content in plaintext HTTP
  - It should be possible to change application description, permissions and/or content using active MiTM and install any malware application! ☺

Remote attack

```
 GET /market/download/Download?
assetId=9177147809749553200&userId=XXXXXXXXXXXXXX&deviceId=YYYYYYYYYYYYYYYYYYY
HTTP/1.1
 Cookie: MarketDA=ZZZZZZZZZZZZZZZZZZZZZZ
 Host: android.clients.google.com
 Connection: Keep-Alive
 User-Agent: AndroidDownloadManager

 HTTP/1.0 200 OK
 ETag: -1625044586
 Content-Type: application/vnd.android.package-archive
 Content-Length: 498162
 Content-Disposition: inline
 Date: Sun, 28 Dec 2010 17:50:13 GMT
 Expires: Sun, 28 Dec 2010 17:50:13 GMT
 Cache-Control: private, max-age=0
 X-Content-Type-Options: nosniff
 X-Frame-Options: SAMEORIGIN
 X-XSS-Protection: 1; mode=block
 Server: GSE
 X-Cache: MISS from proxy
 Via: 1.0 proxy (proxy)
 Connection: keep-alive

 PK.........N.<-...............res/anim/animation_none.xml....].;n.1.E.q.IG."
```

An Android market download

Spying users...

Getting more than location

- Much more interesting information in the different logs:
  - Phone calls (numbers&duration)
  - SMS (PDU format)
- Combination of information:
  - Where did phone calls take place?
  - Where were SMS sent/received?
  - Recovery of deleted SMS, call history...

Getting more than location

- History length?
  - It depends on log filling
    - If user has moved quickly: a few hours
    - If not: nearly a whole day
- Logs size can be changed...

Getting more than location

☞ Complete geolocation, calls and SMS history tracking!

(nearly or no permission needed...)

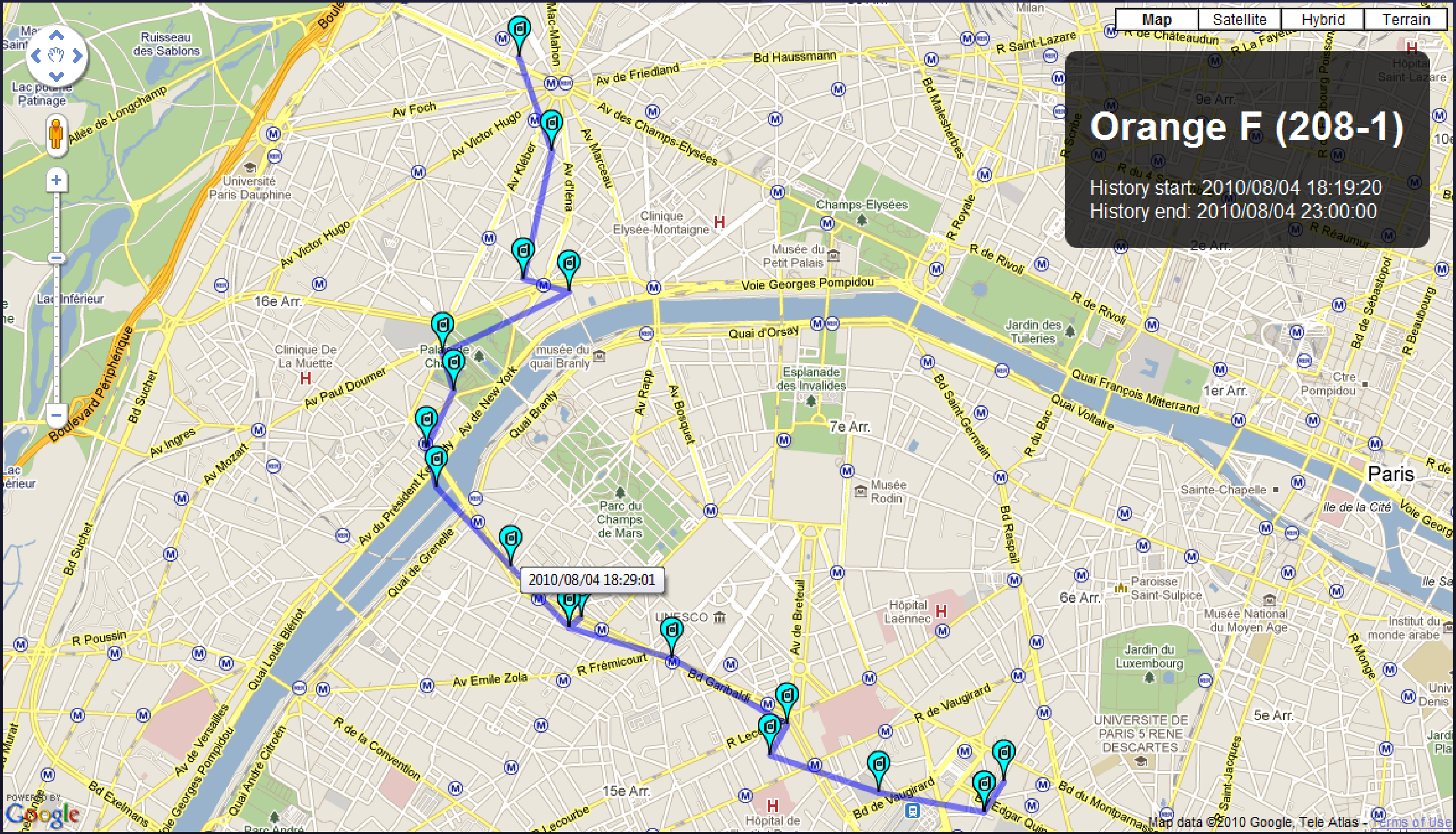
How to protect yourself?

How to protect yourself?

- Carefully look at applications using NDK (apk archives embedding `.so` files)
- Don't install any application requiring READ_LOGS permission
- Don't submit bug reports (or at least choose not to include system logs with submission)
- Reduce logcat buffer size (seems tricky: `logcat -r / logcat -n`)
- Often clear your logcat (`logcat -b radio -c`)
- Disable radio logs (seems tricky too!)

Tool demo

Tool demo

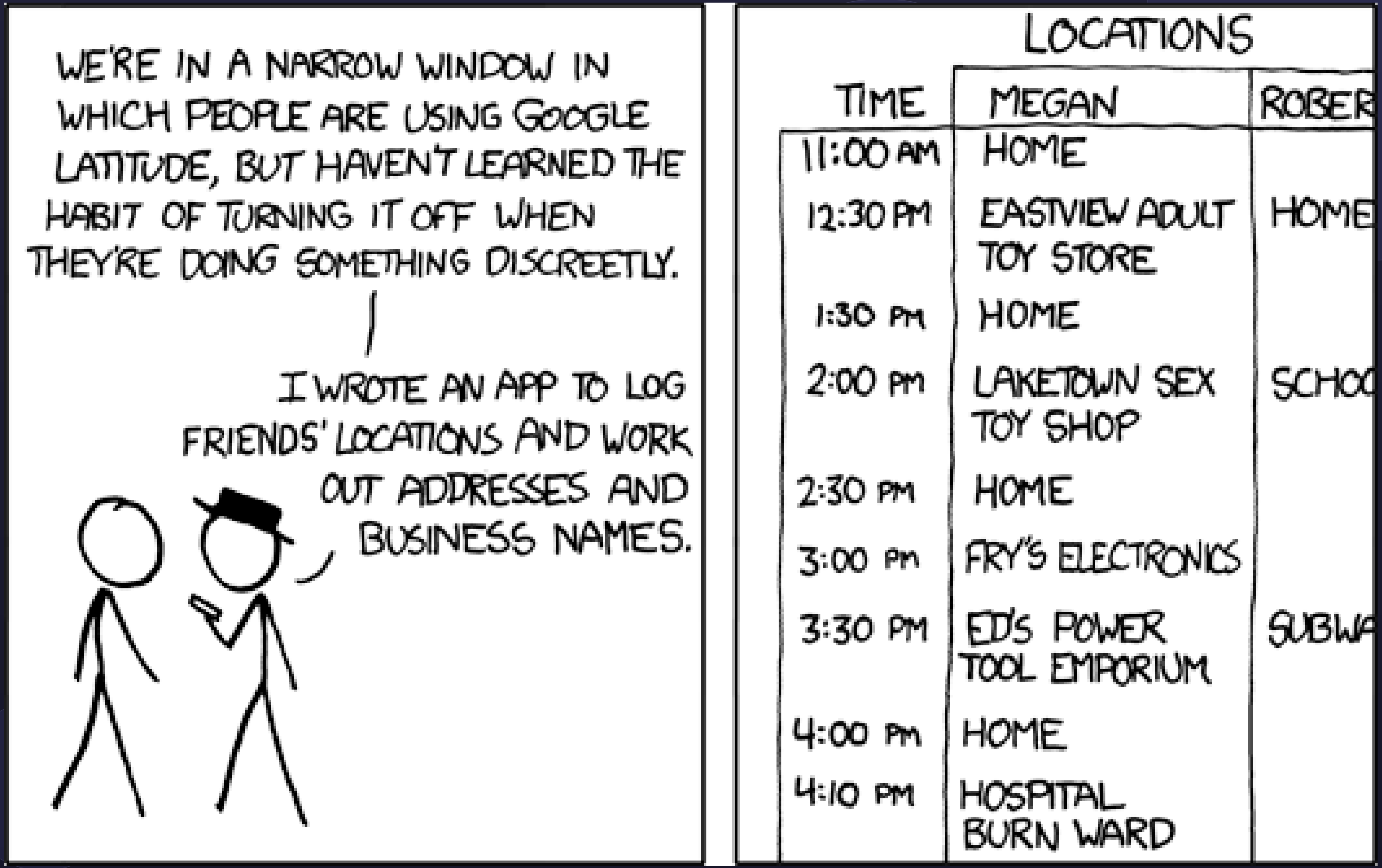


Orange F (208-1)

History start: 2010/08/04 18:19:20
History end: 2010/08/04 23:00:00

2010/08/04 18:29:01

Dumping and viewing a user's past location history


WE'RE IN A NARROW WINDOW IN WHICH PEOPLE ARE USING GOOGLE LATITUDE, BUT HAVEN'T LEARNED THE HABIT OF TURNING IT OFF WHEN THEY'RE DOING SOMETHING DISCREETLY.
I WROTE AN APP TO LOG FRIENDS' LOCATIONS AND WORK OUT ADDRESSES AND BUSINESS NAMES.
LOCATIONS
TIME
MEGAN
ROBER
11:00 AM
HOME
12:30 PM
EASTVIEW ADULT TOY STORE
HOME
1:30 PM
HOME
2:00 PM
LAKETOWN SEX TOY SHOP
SCHOO
2:30 PM
HOME
3:00 PM
FRY'S ELECTRONICS
3:30 PM
ED'S POWER TOOL EMPORIUM
SUBWA
4:00 PM
HOME
4:10 PM
HOSPITAL BURN WARD

Comic by http://xkcd.com
Licensed under the CC Attribution-NonCommercial 2.5 Generic license

That's all folks!

Hope you enjoyed the talk!

Any questions?

Many thanks for attending!