iPhone Privacy

Nicolas Seriot

Black Hat DC 2010

Arlington, Virginia, USA

http://seriot.ch

Twitter @nst021

Who am I?

- **Nicolas Seriot**, Switzerland
- HES Software Engineer
- Cocoa developer and **iPhone programming trainer** at Sen:te

- Data-mining research assistant at Swiss University of Applied Sciences (HEIG-VD) since 2009
- **MAS in Economic crime investigation**

You said... Switzerland?

Outline

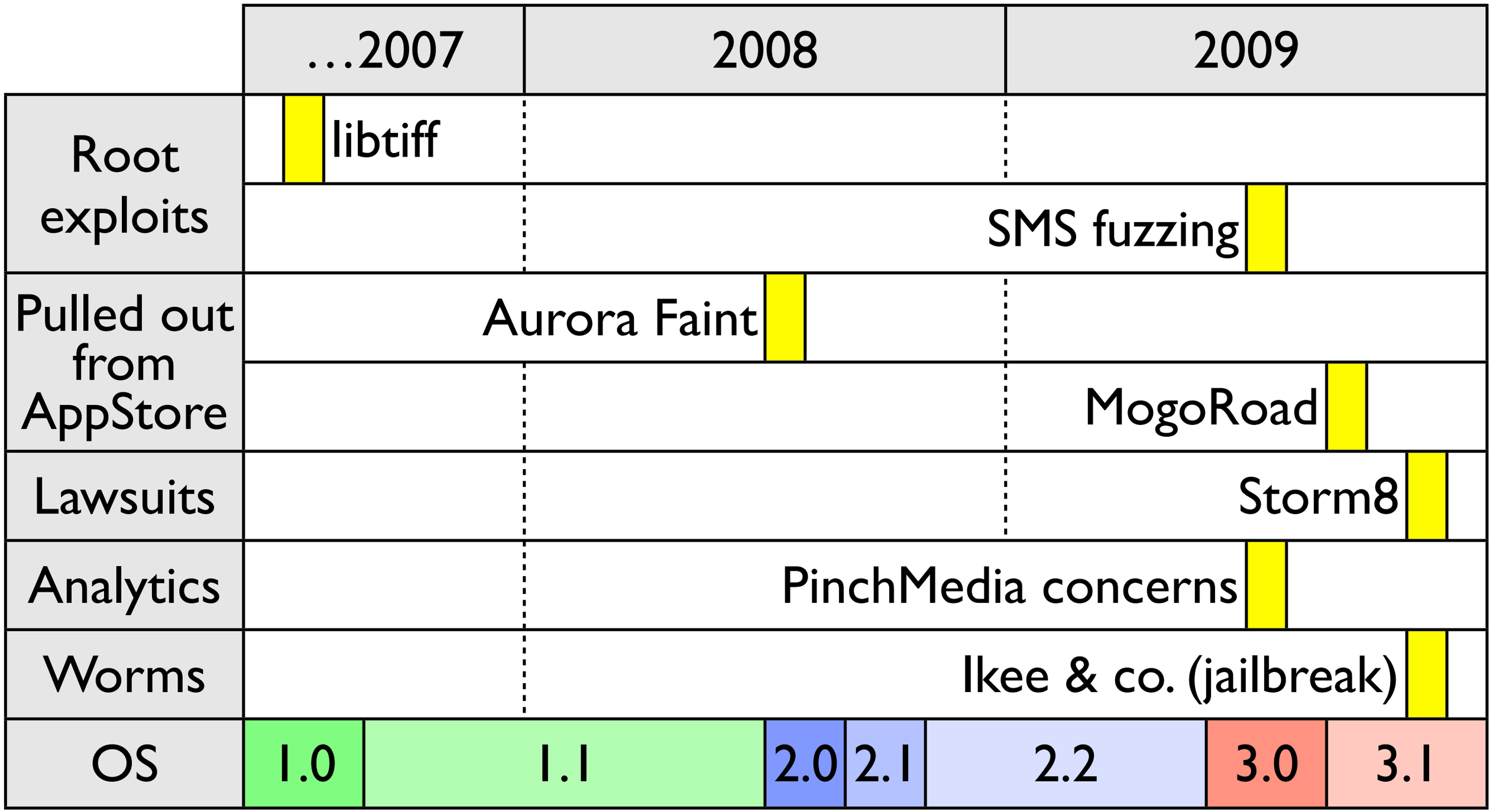
1. **Privacy issues overview**
2. **What can iPhone spyware do?**
    1. Access personal data
    2. Fool App Store's reviewers
3. **Attack scenarios**
4. **Recommendations and conclusion**

iPhone Catch Up

- **iPhone**
  - **34 millions devices** worldwide
- **Apple's App Store**
  - 140,000 applications, **3 billion downloads**
- **Jailbreak**
  - non-official firmwares, will also run unsigned code, often installed with sshd

I. Privacy Issues Overview

Privacy Issues Timeline



Root Exploits

- **libtiff** – July 2007
  - Multiple buffer overflows by Tavis Ormandy, exploited by Rik Farrow
    - Patched in iPhone OS 1.1.2
- **SMS fuzzing** – July 2009
  - Demonstrated at Black Hat USA 2009 by Charlie Miller and Collin Mulliner
  - Patched in iPhone OS 3.0.1

Root Exploits

TUESDAY, FEBRUARY 02, 2010

iPhone OS and Mac OS X Stack Buffer Overflow

My second security advisory in 2010 (TKADV2010-002) describes the details of a stack buffer overflow I found in CoreAudio of Apple's iPhone OS and Mac OS X. The bug can be triggered by playing a maliciously crafted mp4 audio file. Example attack vectors on the iPhone are MobileSafari and malicious ringtones.

Crashdump details:

```
[..]
Process:         mediaserverd [17]
Path:            /usr/sbin/mediaserverd

..

Exception Type:  EXC_BAD_ACCESS (SIGSEGV)
Exception Codes: KERN_INVALID_ADDRESS at 0x41414140

..

Unknown thread crashed with ARM Thread State:
    r0: 0x6474613f    r1: 0x01380c40      r2: 0x380c561c      r3: 0x0000010d
    r4: 0x41414141    r5: 0x41414141      r6: 0x41414141      r7: 0x41414141
    r8: 0x41414141    r9: 0x00181494     r10: 0x41414141     r11: 0x41414141
    ip: 0x00818000    sp: 0x01380c00      lr: 0x3072d454      pc: 0x41414140
  cpsr: 0x60000030
[..]
```

POSTED BY TK AT 10:01 PM

http://tk-blog.blogspot.com/2010/02/iphone-os-and-mac-os-x-stack-buffer.html

Analytics Frameworks

- **PinchMedia**
  - Think Google Analytics for your app
  - July 2009 – bloggers raise privacy concerns
  - Users are not informed and can’t opt-out

Create your own Trusted Certificate!

February 2, 2010, 1:04PM

iPhones Vulnerable to New Remote Attack

by Dennis Fisher

Share Recommend (2) Print E-mail

2 Comments

There are several flaws in the way that the iPhone handles digital certificates which could lead to an attacker being able to create his own trusted certificate and entice users into downloading malicious files onto their iPhones. The attack is the end result of a number of different problems with the way that the iPhone handles over-the-air provisioning, trusted root certificates and configuration files. But the result of the attack is that a remote hacker may be able to change some settings on the iPhone and force all of the user's Web traffic to run through any server he chose and also to change the root certificate on the phone, enabling him to man-in-the-middle SSL traffic from the iPhone.

http://threatpost.com/en_us/blogs/iphones-vulnerable-new-remote-attack-020210

Storm8 Lawsuit

Backdoor in top iPhone games stole user data, suit claims

Storm8's iSpy

By **Dan Goodin in San Francisco** • **Get more from this author**

Posted in Mobile, 6th November 2009 06:02 GMT

A maker of some of the most popular games for the iPhone has been surreptitiously collecting users' cell numbers without their permission, according to a federal lawsuit filed Wednesday.

The complaint claims best-selling games made by Storm8 contained secret code that bypassed safeguards built into the iPhone to prevent the unauthorized snooping of user information. The Redwood City, California, company, which claims its games have been downloaded more than 20 million times, has no need to collect the numbers.

"Nonetheless, Storm8 makes use of the 'backdoor' method to access, collect, and transmit the wireless phone numbers of the iPhones on which its games are installed," states the complaint, which was filed in US District Court in Northern California. "Storm8 does so or has done so in all of its games."

Messages left for Storm8 representatives weren't returned.

The Register®
Biting the hand that feeds IT

http://www.theregister.co.uk/2009/11/06/iphone_games_storm8_lawsuit/

http://www.boingboing.net/lawsuits/Complaint_Storm_8_Nov_04_2009.pdf

Pulled out from AppStore*

- **Aurora Feint** – July 2008
  - Sent contact emails in clear
  - 20 million downloads
- **MogoRoad** – September 2009
  - Sent phone number in clear
  - Customers got commercial calls

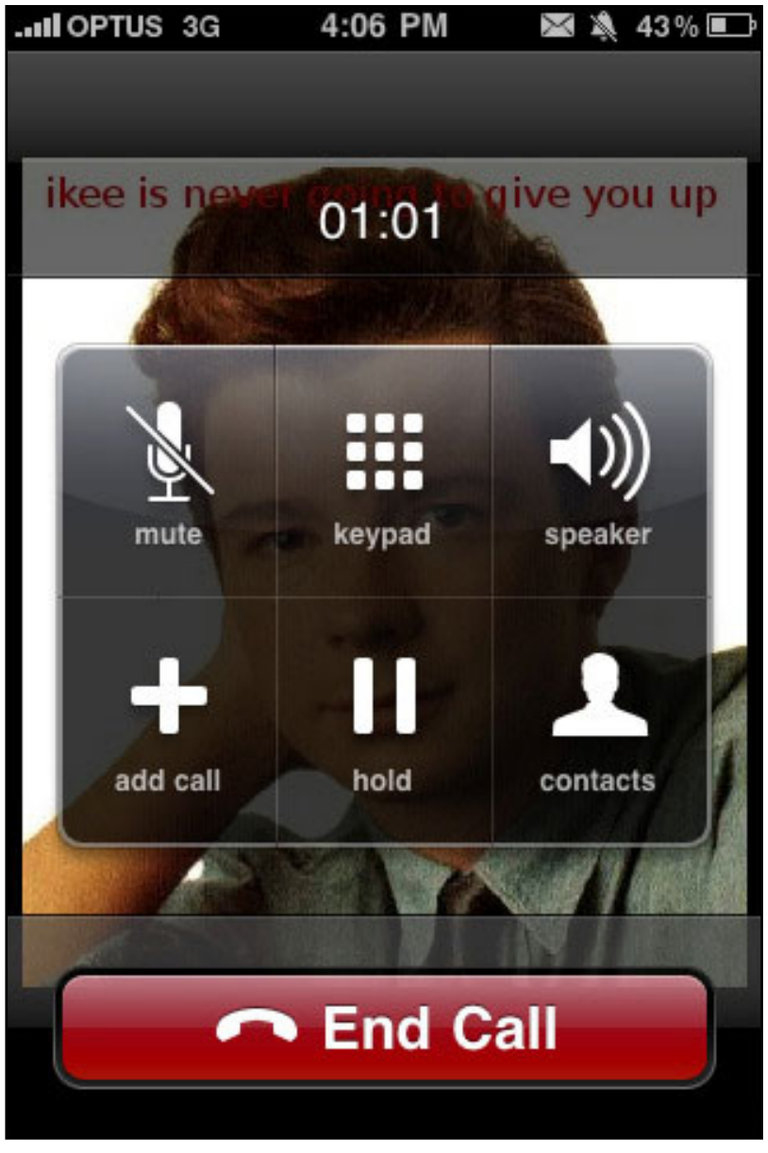
* Both applications are back on AppStore after updating their privacy policy.

2009-11 Worms / Jailbreak

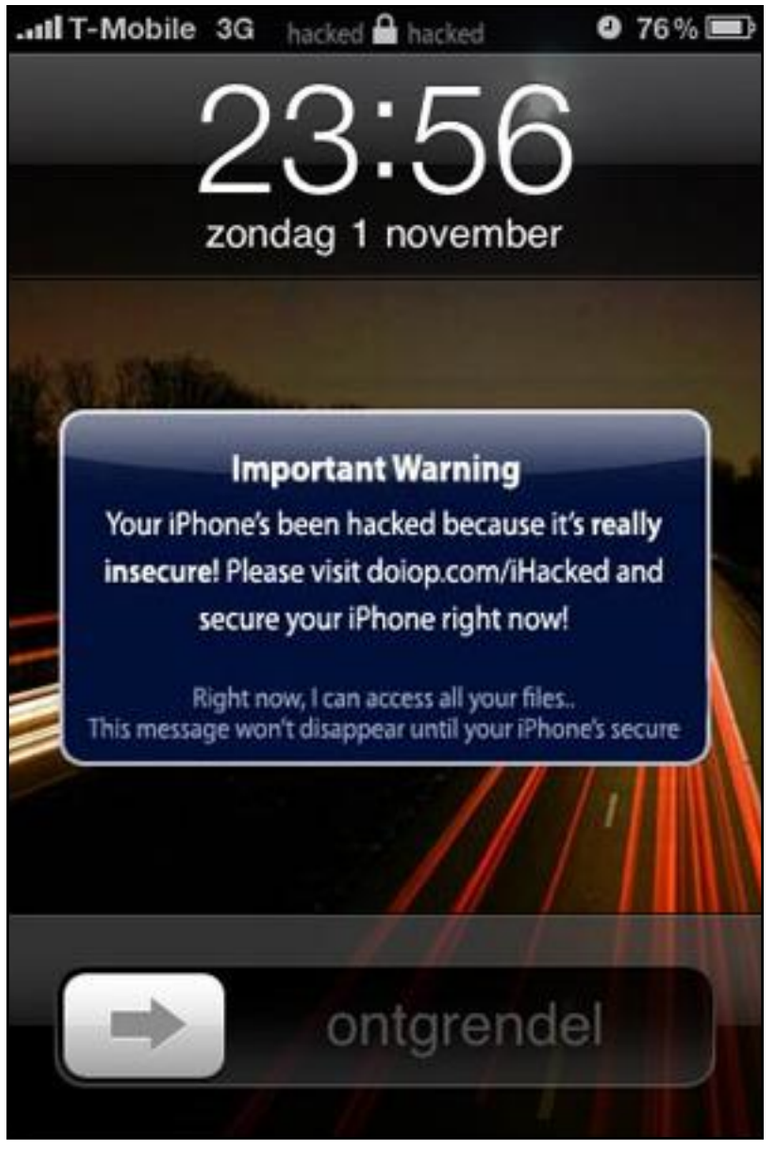
- Exploiting default root password on SSH

1. **Ikee** – changes wallpaper to Rick Astley
2. **Dutch 5 € ransom** – locks iPhone against a ransom (not refunded)
3. **IPhone/Privacy.A** – steals iPhone content, invisible, no replication
4. **Duh / Ikee.B** – steals iPhone content, changes root password, Lithuanian botnet (analysis)

This is what it looks like



Ikee



Dutch 5 € ransom

Apple Gets Bad Press

SOPHOS

This further demonstrates that iPhones are **not ready** for the business environment.

http://www.sophos.com/blogs/chetw/g/2009/11/21/malicious-iphone-worm-loose/

IMHO, this is not more clever as claiming that Linux is not ready for business since you can exploit a weak default root password on SSH…

2. What can iPhone Spyware do?

Technical Context

- Imagine **a rogue breakout** on AppStore
- iPhone OS version 3.1.3
  - **No jailbreak** (no root access, 6-8 % iPhones)
  - No hardware attacks (don't lose your iPhone)
  - **Not calls to private APIs** (there's no need to)
  - No Facebook or Twitter profile data…
  - No root shells exploits
- Look for entry points, look for **personal data**

Methodology – Step A

Access
personal data

2.1. Access Personal Data

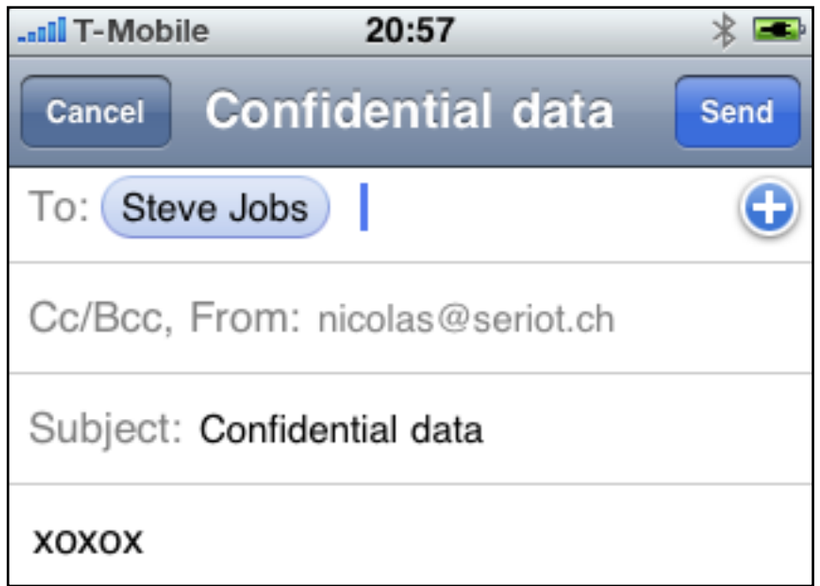
Cell Numbers

```
NSDictionary *d =
    [NSUserDefaults standardUserDefaults];
NSString *phone =
    [d valueForKey:@"SBFormattedPhoneNumber"];
```

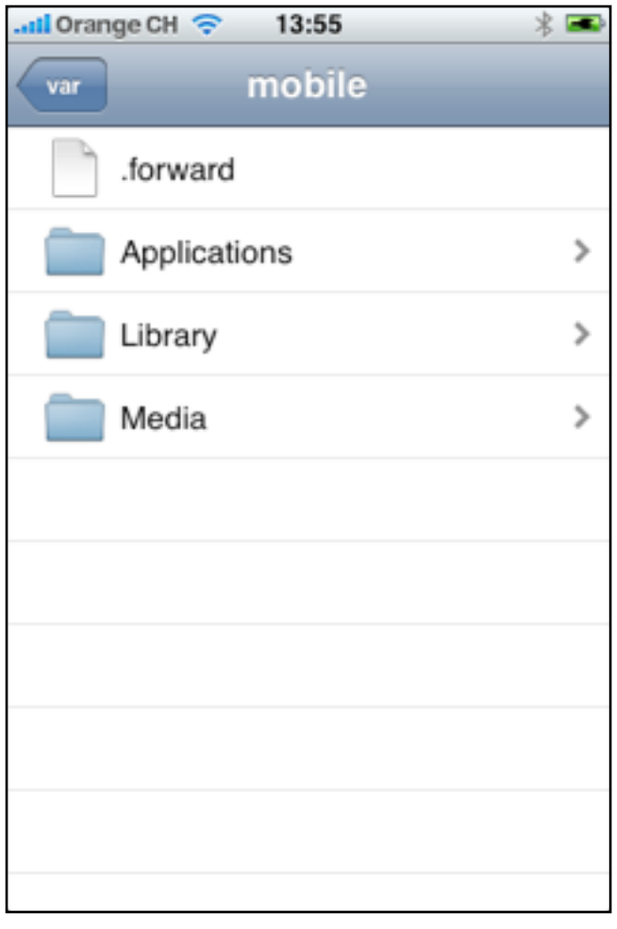
- Entered in iTunes
- Optional, you can safely change it

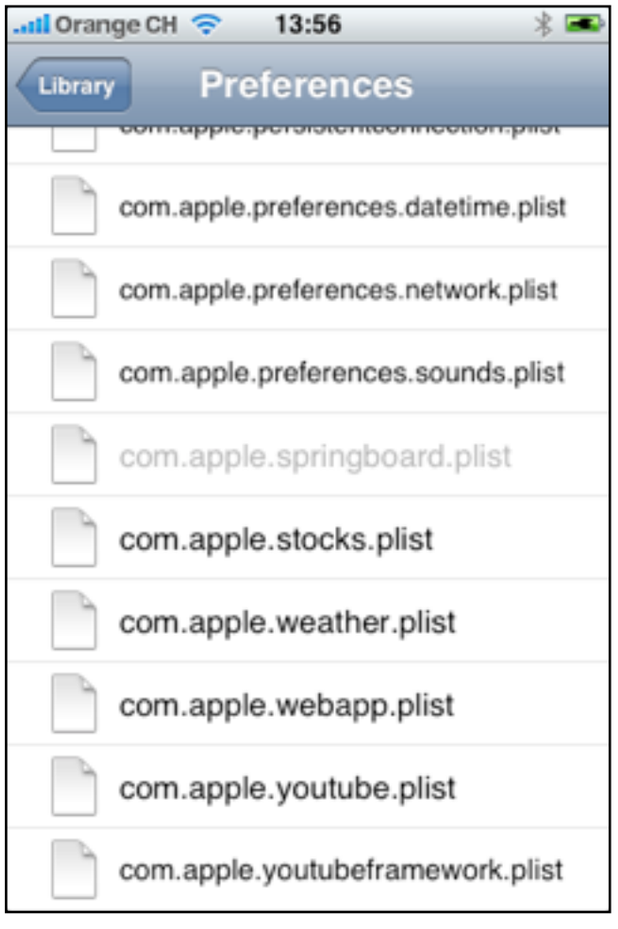
Address Book API

- No “Me” record
- Unrestricted read/write access
- Tampering with data
- change `*@ubs.com` into `pirate123@gmail.com`



File System Access





http://fswalker.googlecode.com

iPhone Sandboxing

- Restricts applications access to OS resources
- A list of deny/allow rules at kernel level
- `/usr/share/sandbox/SandboxTemplate.sb`

```
(version 1)
(deny default)

; Sandbox violations get logged to syslog
via kernel logging.
(debug deny)

(allow sysctl-read)

; Mount / umount commands
(deny file-write-mount file-write-umount)
```

```
; System is read only
(allow file-read*)
(deny  file-write*)

; Private areas
(deny  file-write*
       (regex "^/private/var/mobile/
Applications/.*$"))
(deny  file-read*
       (regex "^/private/var/mobile/
Applications/.*$"))
```

Sandboxing for the Win?

Applications on the device are "**sandboxed**" so they **cannot access data stored by other applications**.

In addition, **system files, resources**, and the kernel **are shielded from the user's application space**.

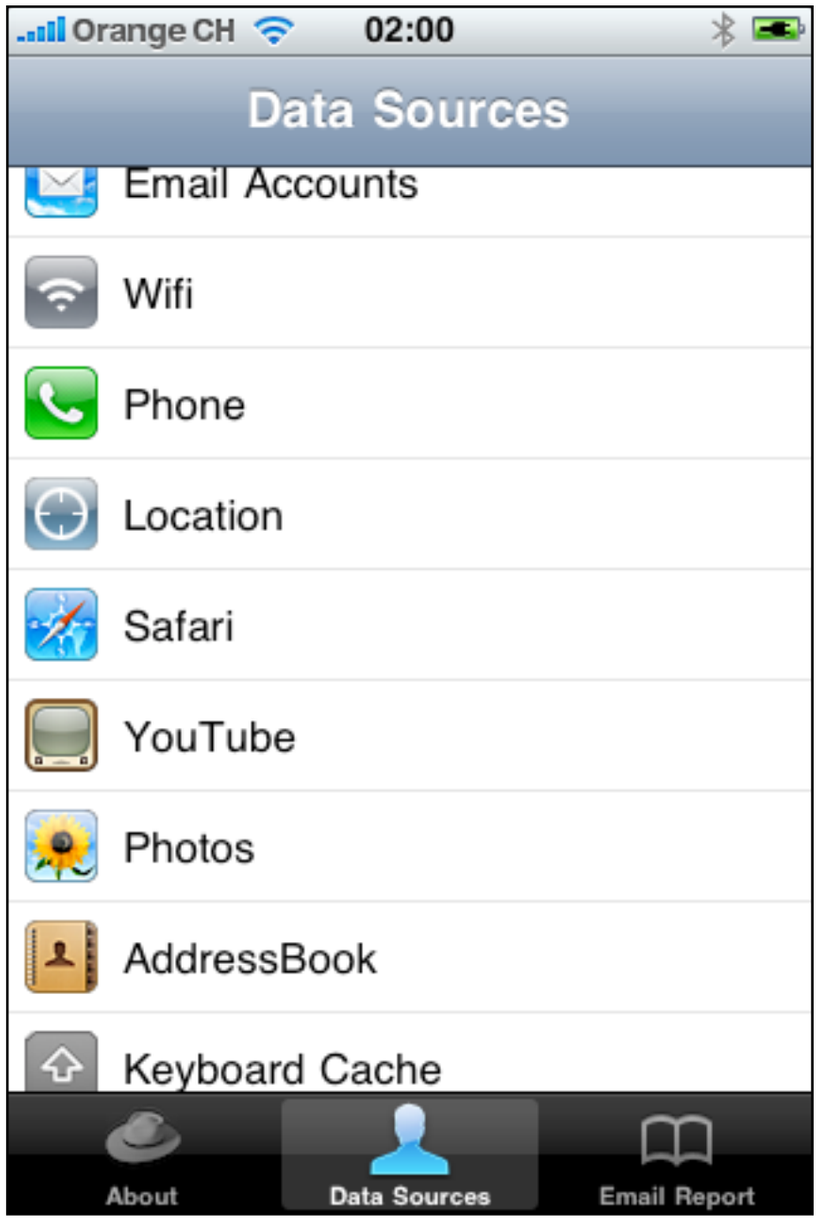
Apple – iPhone in Business – Security Overview

http://images.apple.com/iphone/business/docs/iPhone_Security_Overview.pdf

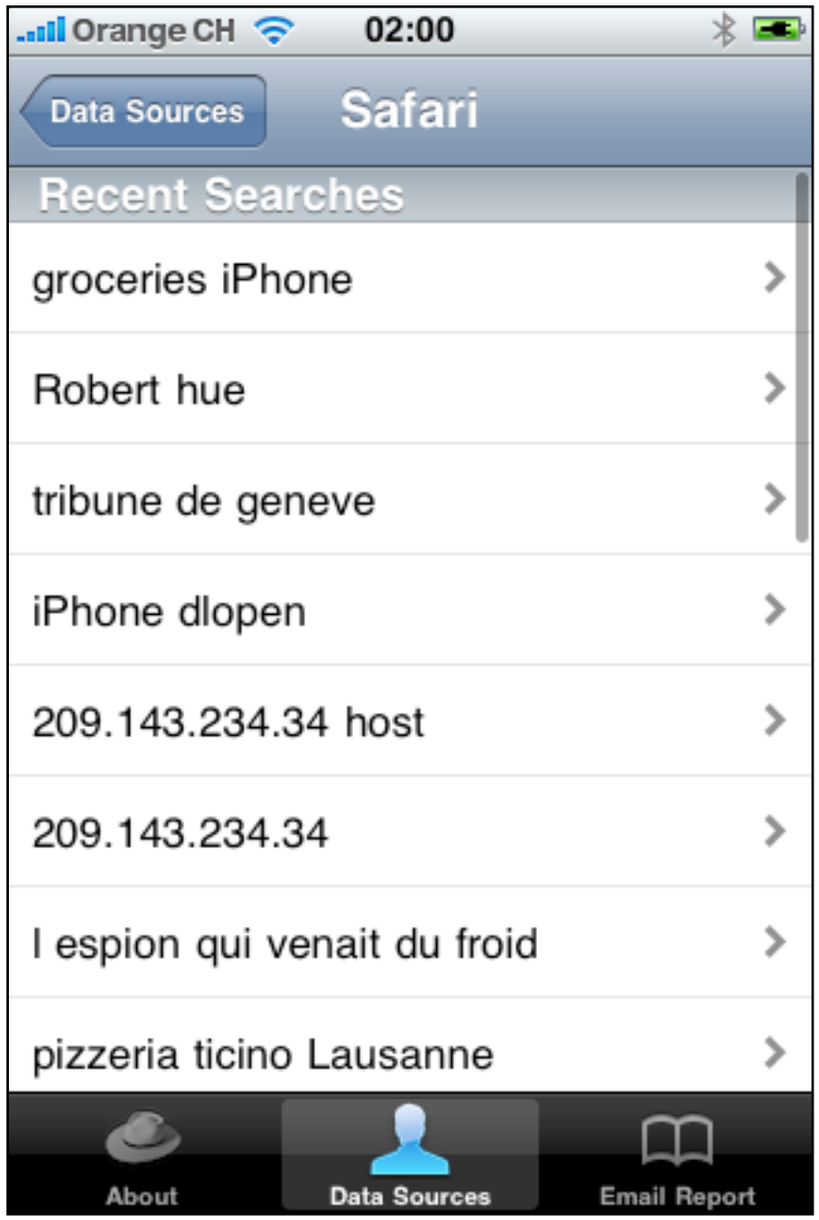
This is not true, because rules are too loose.

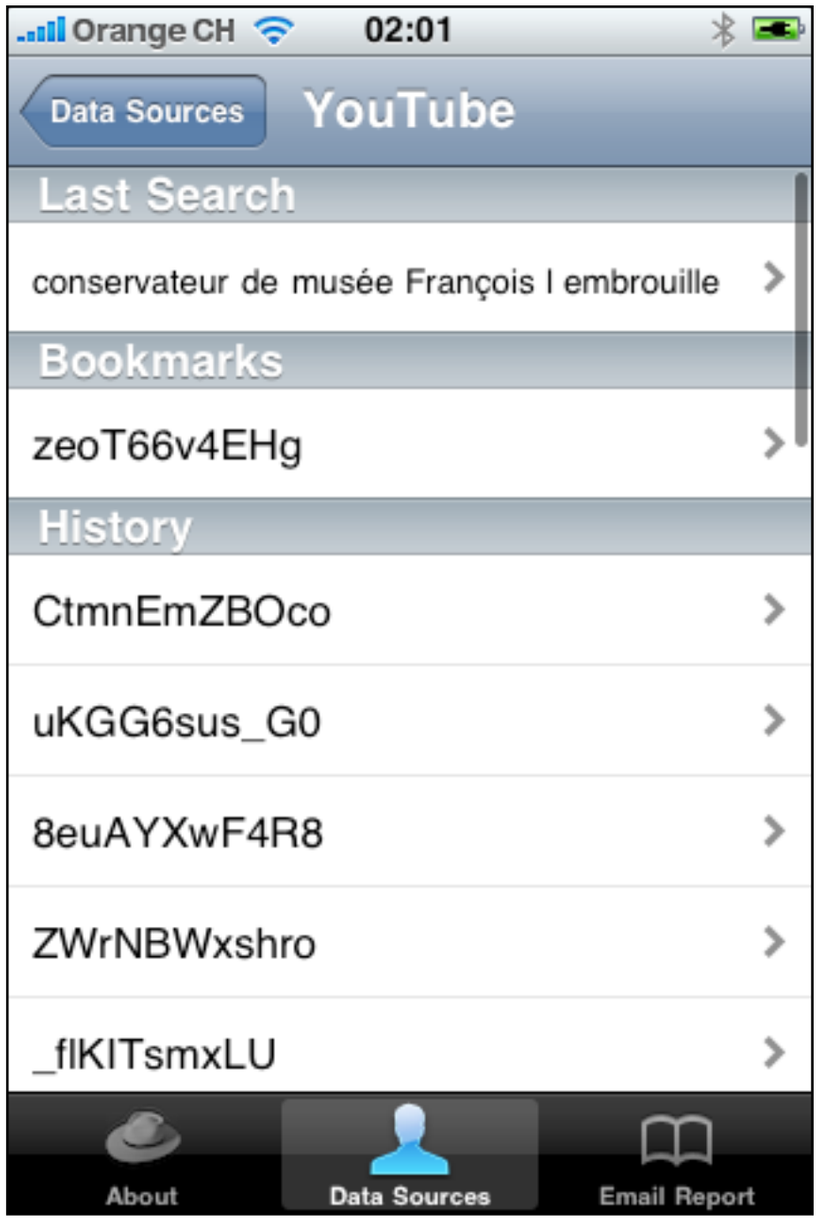
Demo!

Introducing SpyPhone

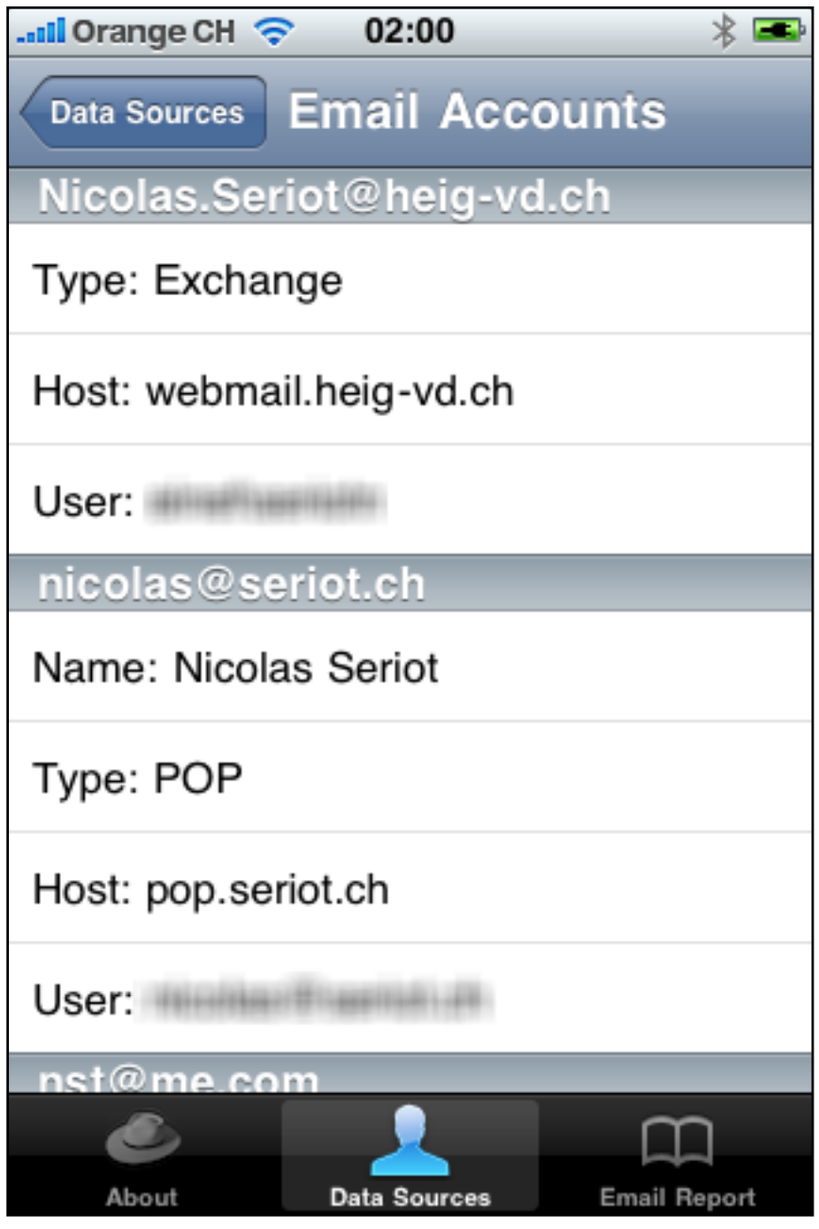


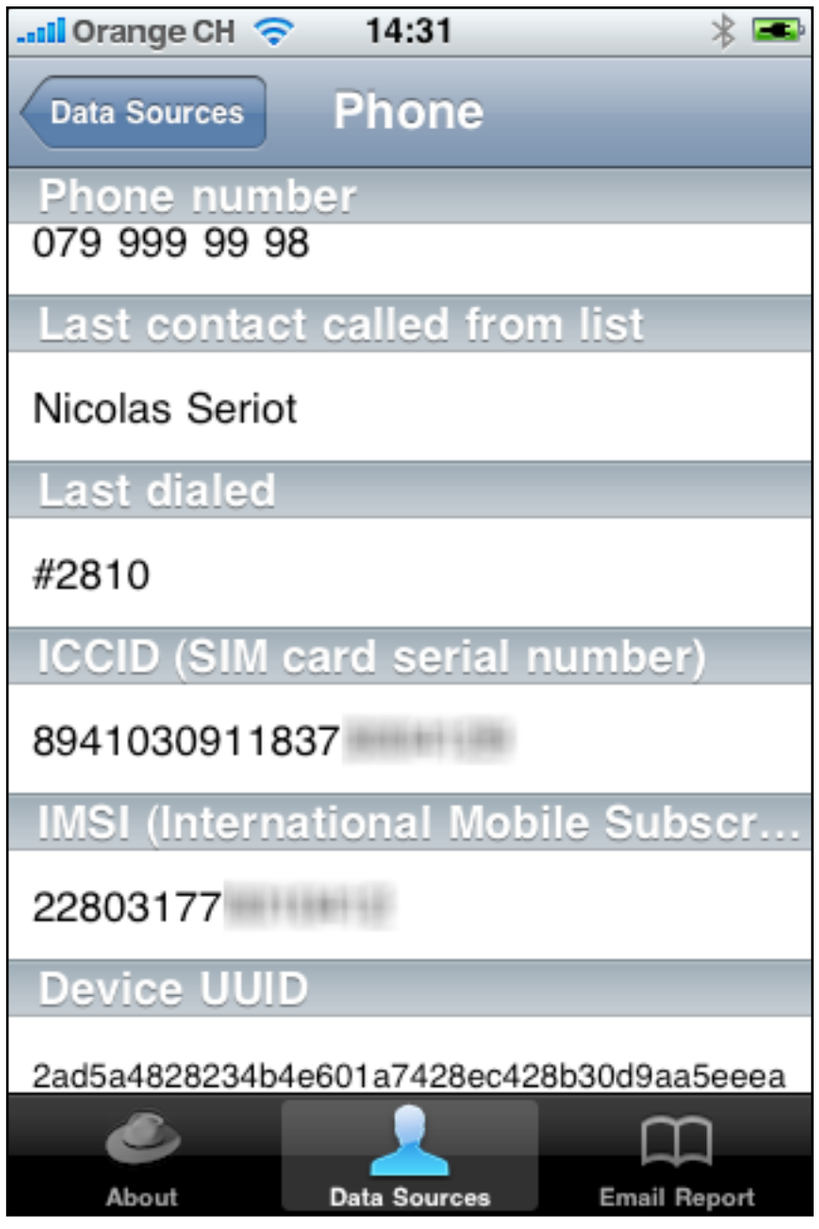
Safari / YouTube Searches



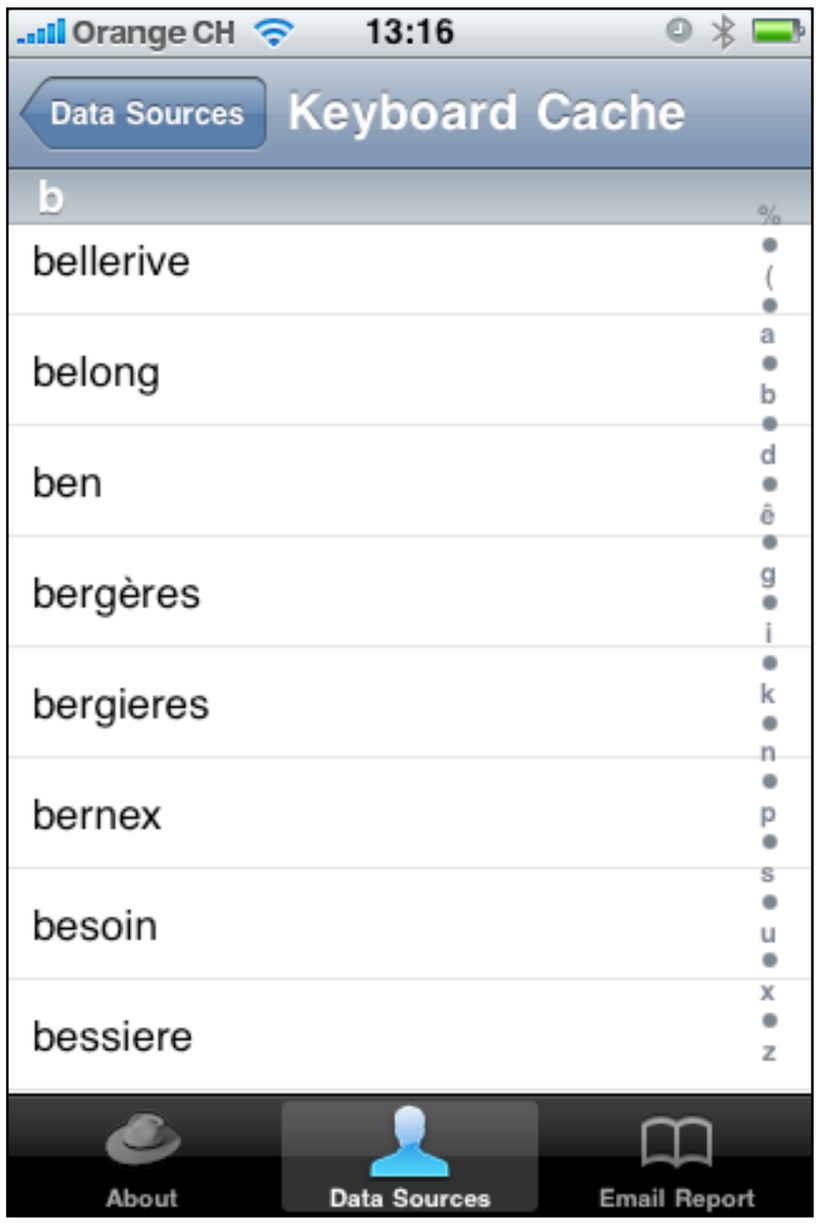


Phone and Email Accounts

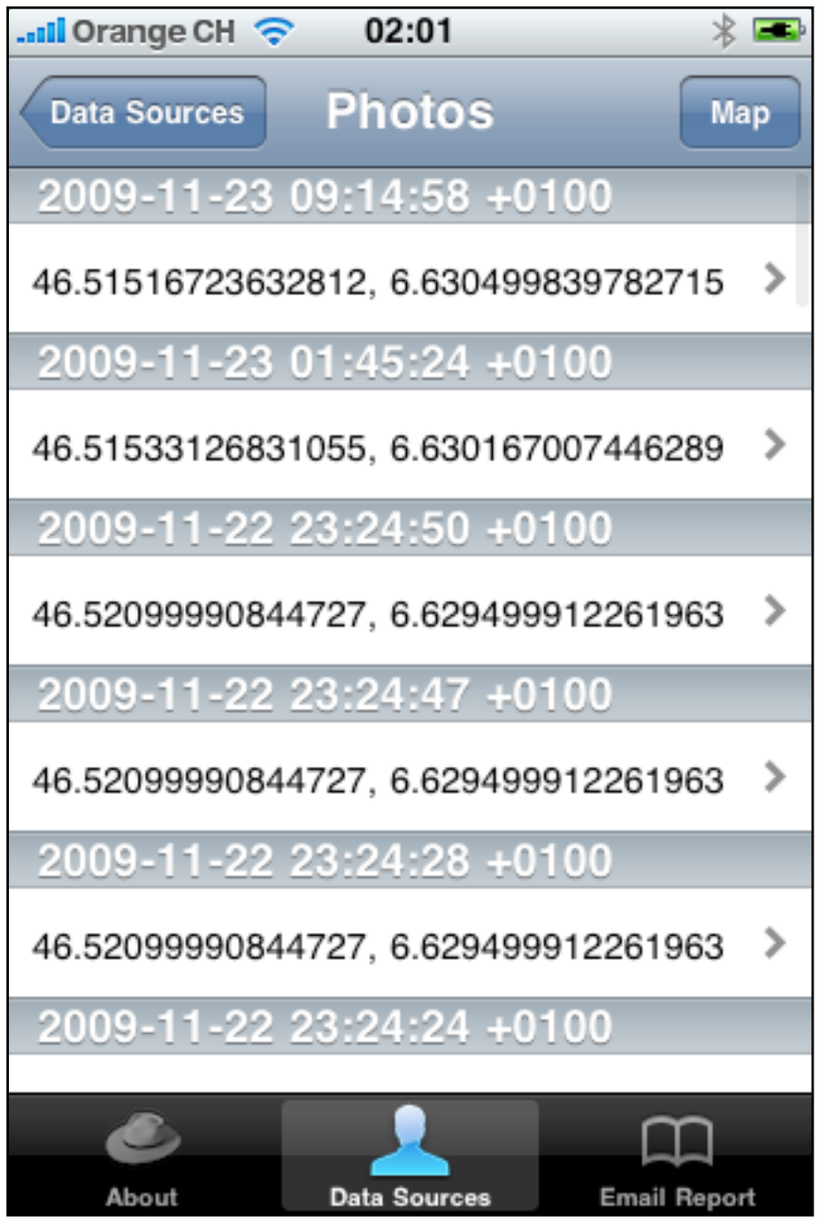




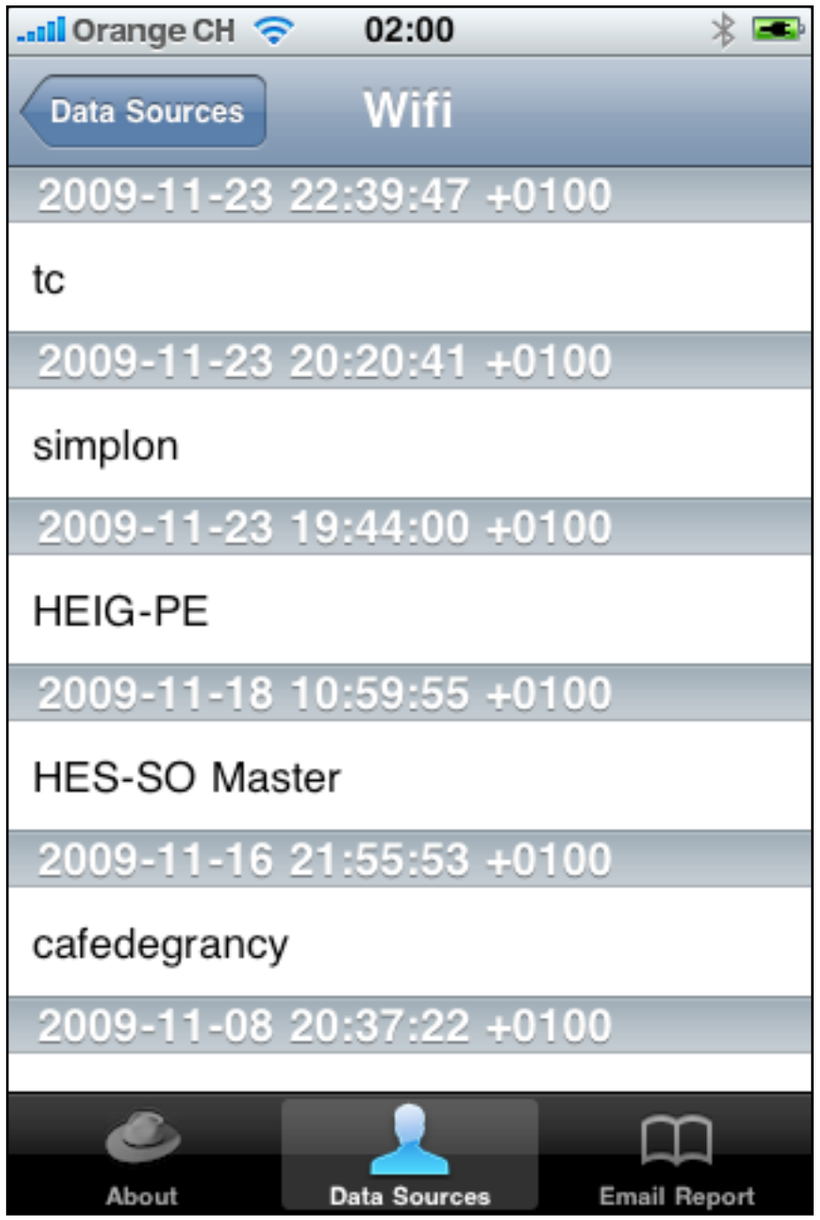
Contacts, Keyboard Cache



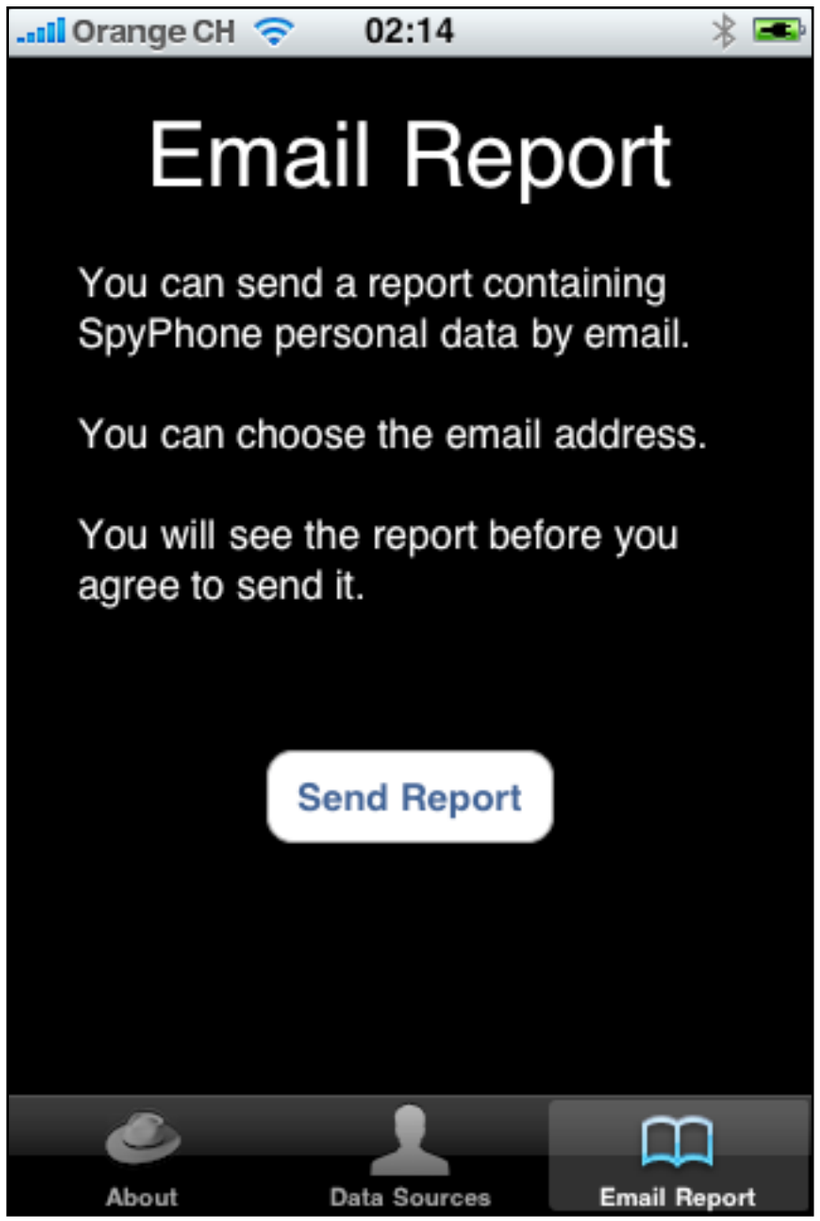
Geotagged Photos Location



GPS and Wifi Location

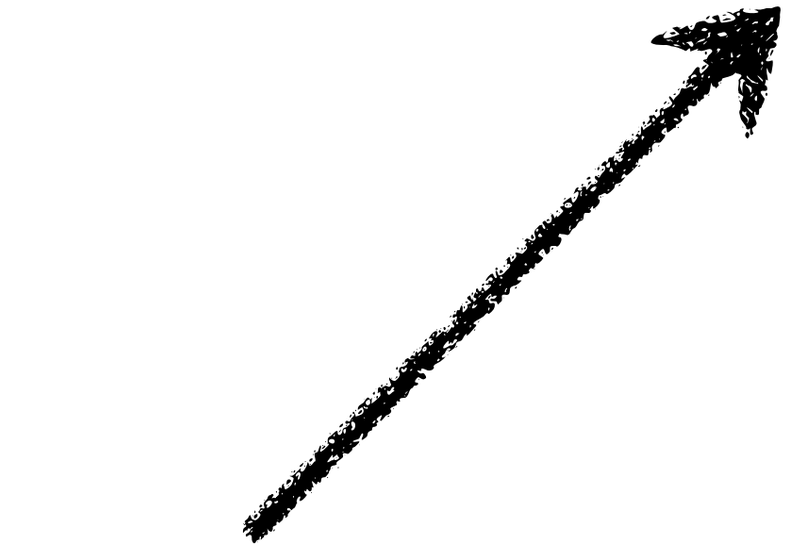


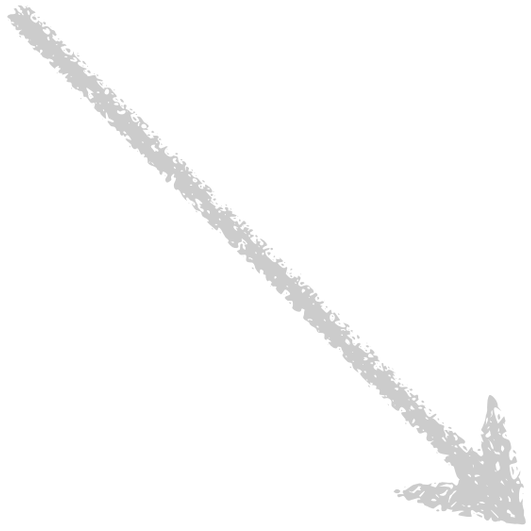
SpyPhone



- Contributions welcome!
- 2000 lines + EXIF library
- GPL License
- http://github.com/nst/spyphone

Methodology – Step B





Put the application
on the App Store.

2.2. Fool App Store Reviewers

App Store and Malware

We've built **a store** for the most part **that people can trust**.

There have been applications submitted for approval that will **steal personal data**.

- Phil Schiller, Apple senior VP

http://www.businessweek.com/technology/content/nov2009/tc20091120_354597.htm

10,000 submissions per week
10% of rejections related to malware

iPhone SDK Standard Agreement

- **5.4 – You may not make any public statements** regarding this Agreement
  - Applications must not collect users' personal information and must comply with local laws
  - Base for spyware rejection
  - Published by WikiLeaks and Wired…

AppStore Reviews

- **Reviewers can be fooled**
  - Spyware activation can be **delayed**
  - Payloads can be **encrypted**
  - Many things can **change at runtime**

Hiding the Beast

- **Guesswork** about AppStore review process
- **Static analysis** with `$ strings`
- **Dynamic analysis** with I/O Instruments
  - Monitor file openings
- Check against black lists

Strings Obfuscation

```
- (NSString *)stringMinus1:(NSString *)s {
  NSMutableString *s2 = [NSMutableString string];
  for(int i = 0; i < [s length]; i++) {
      unichar c = [s characterAtIndex:i];
      [s2 appendFormat:@"%C", c-1];
  }
  return s2;
}

- (void)viewDidAppear:(BOOL)animated {
    NSString *pathPlus1 =
        @"0wbs0npcjmf0Mjcsbsz0Qsfgfsfodft0dpn/bqqmf/bddpvoutfuujoht/qmjtu";
     // @"/var/mobile/Library/Preferences/com.apple.accountsettings.plist"
    NSString *path = [self stringMinus1:pathPlus1];
    NSDictionary *d = [NSDictionary dictionaryWithContentsOfFile:path];
    // ...
}
```

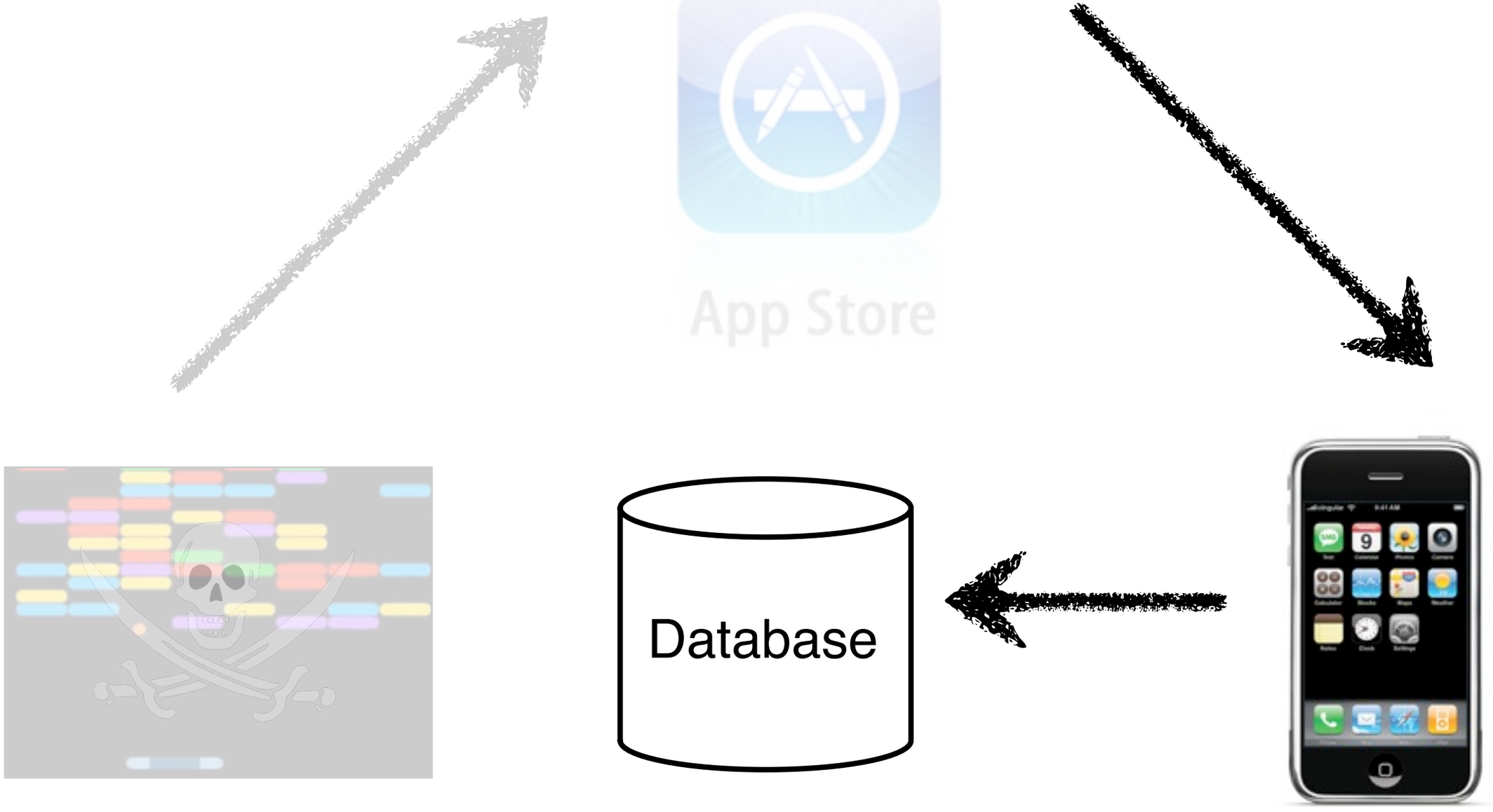
This code would probably pass a static analysis

Apple's GPS Kill Switch

```
$ curl https://iphone-services.apple.com/clbl/unauthorizedApps
{
    "Date Generated" = "2010-01-03 05:02:36 Etc/GMT";
    "BlackListedApps" = {};
}
```

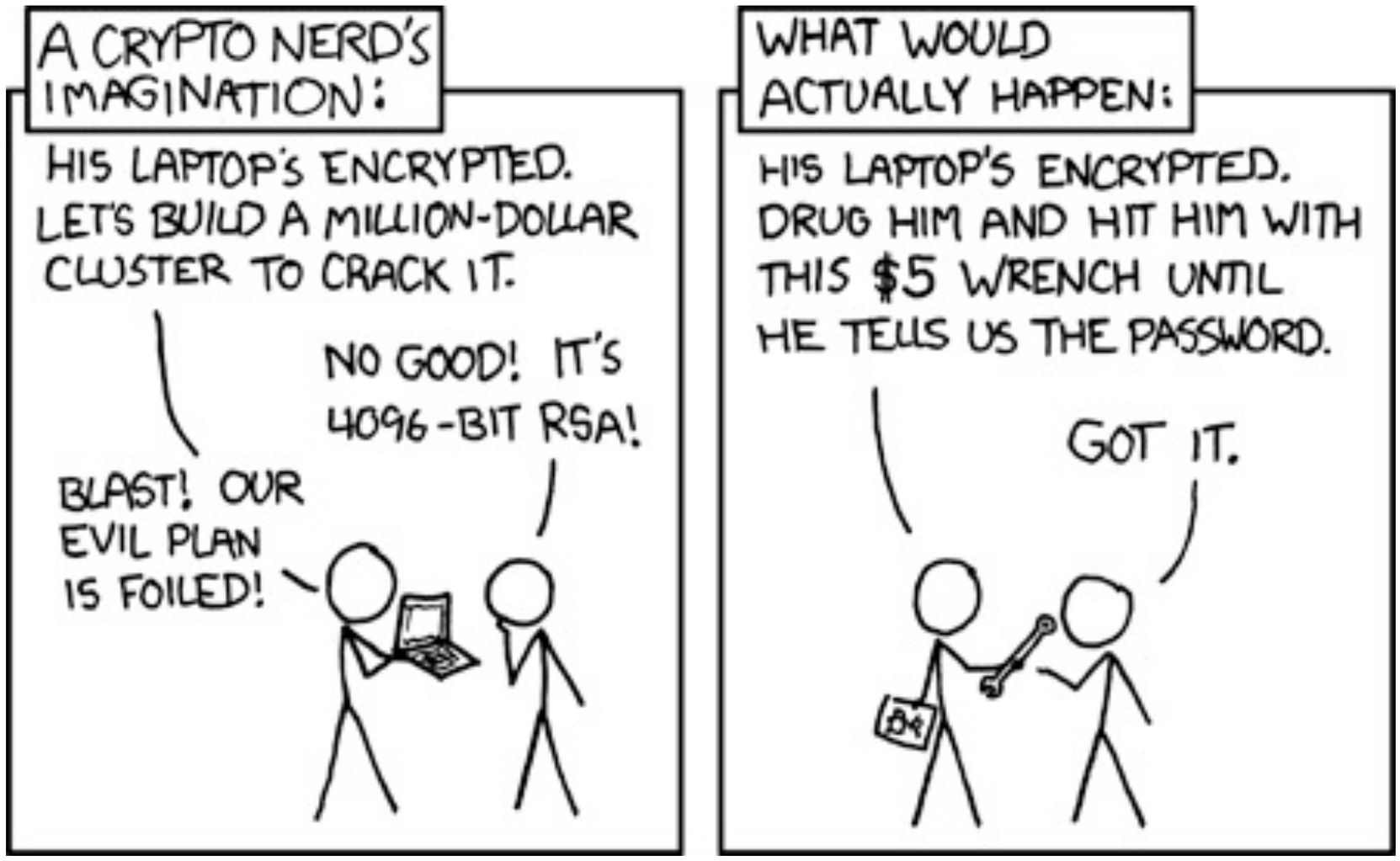
- Discovered by Jonathan Zdziarski in August 2008
- clbl stands for "Core Location Black List"
- Prevent applications from using Core Location
- Apple never acknowledged its existence publicly
- Apple never used it – SpyPhone doesn't care

Methodology – Step C



4. Attack Scenarios

This is Real World



http://xkcd.com/538/

The Spammer

- Write a little **breakout game**
- Make it available for free on AppStore
- Collect user **email addresses** + **weather cities** + **user's interests** from Safari searches and keyboard cache
- Collect Address Book emails
- Send them with high scores

The Luxury Products Thief

- Write an app for sports car or luxury watches collectors
- Report the name, phone, area and geotagged photos of healthy people
- When you can determine that someone is away from home, just rob him

The Jealous Husband

- Could also be named **evil competitor** or **law enforcement officer**
- Requirements: **5 minute** physical access to the device, an Apple $99 developer license, a USB cable
- **Install SpyPhone, send the report**
- Delete the report from sent emails, delete SpyPhone

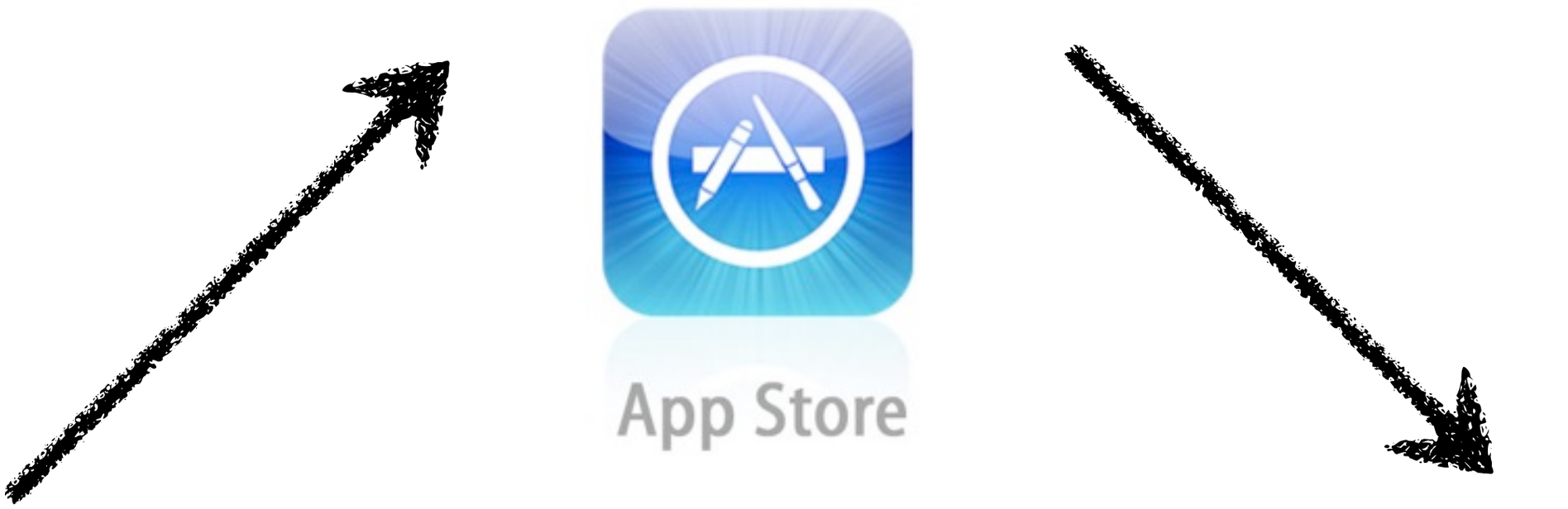
http://www.flickr.com/photos/11213613@N05/4147756184/

VIPs

François Fillon, French Prime Minister, and
Rachida Dati, former Justice French Minister

< insert your attack scenario here >

Methodology



Database

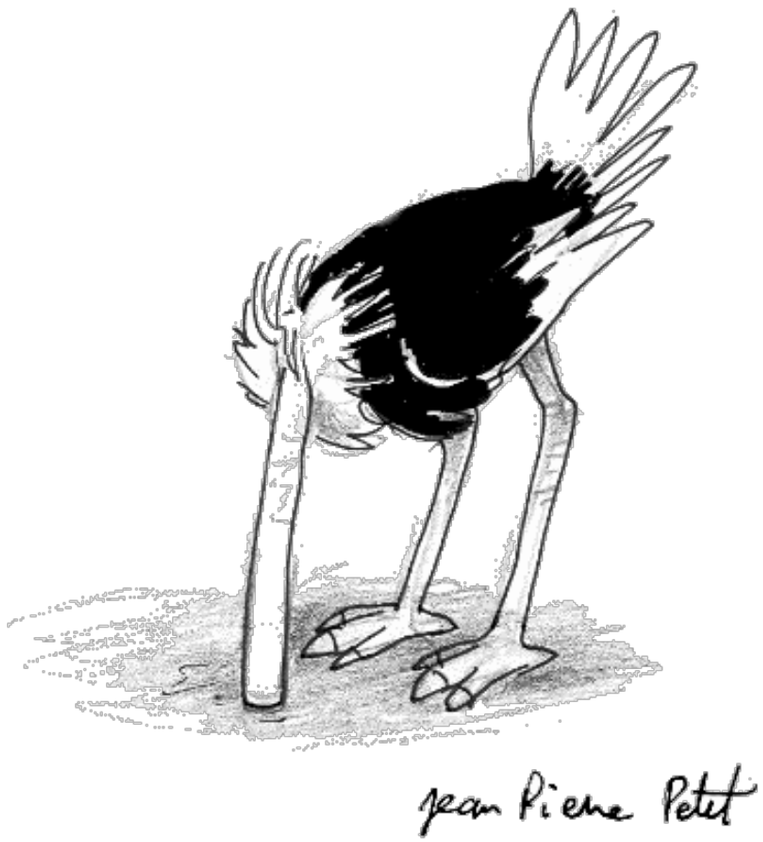
So what?

4.

Recommendations and Conclusion

Security Through Obscurity

- **Apple** should not rely on security through obscurity
- It shouldn't claim that an application cannot access data from other applications
- It may have to **review the iPhone S-SDLC**



Keyboard, Firewall, …

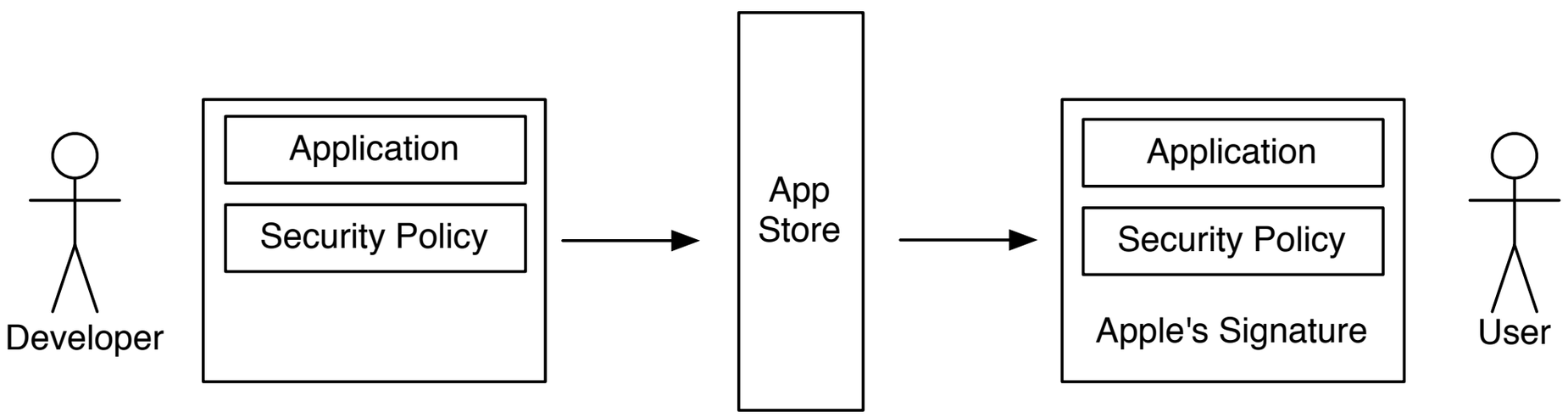
- Clearly, the Keyboard cache shouldn't be readable, it should be a system service instead
- Something like an **applicative firewall** should inform the user and let him prevent access
- A network firewall should also be available to let the user **opt-out** from the various analytics frameworks

Address Book

- Users should be required to grant **read**-access to the Address Book, as for the GPS location
- Users should be prompted again if the application attempts to **edit** the Address Book
- Risk: **being overwhelmed with pop-ups**

Toward Apple approved Security Policies?

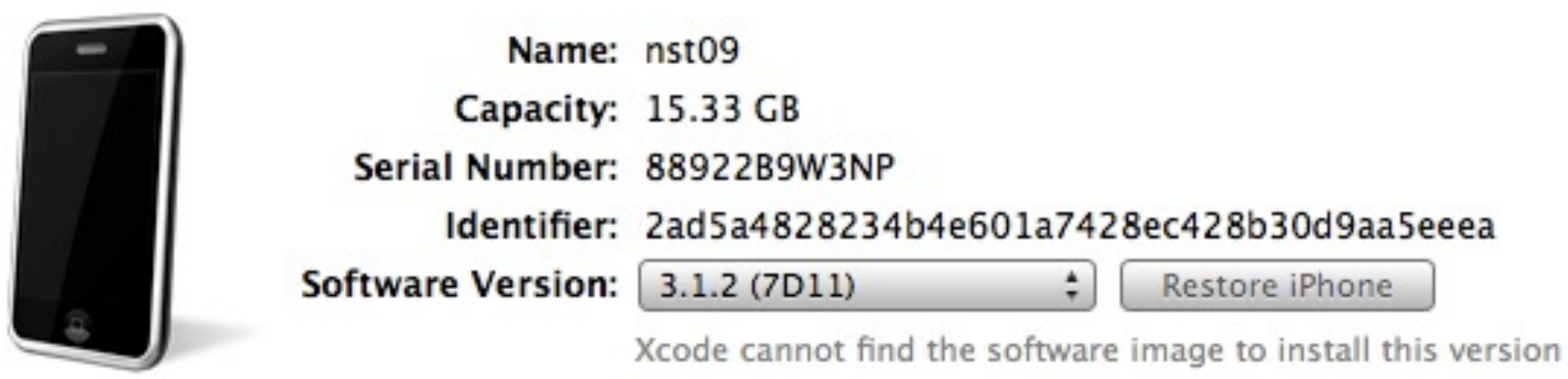
Apple could ask developers to establish a security policy, stating what the application can do.



eg. read the AddressBook but not elsewhere on the file system, access the Internet but not the GPS

Device Unique Identifiers

- The user should be prompted when an application attempts to access the UUID
- UUID may be used to link data gathered by different applications and frameworks
- Apple should introduce an **app-device identifier**, unique for (device, application)‘



Okay, but…

is there anything can do?

Consumers

- Beware of the application they install
- Use common sense
- **Remove their cell number** from Settings
- **Reset** keyboard and Safari **caches regularly**

Professionals

- **Assess risks** correctly, especially if they are **required by law to keep secrets**.
  - Medical staff, bankers, attorney, law enforcement officers…
- **Use Apple's program for enterprise deployment**, which lets administrators define profiles that enforce restrictions.

Conclusion

- Assume that spyware **are** on the AppStore
  - 1$ ecosystem doesn't help
  - **Massive privacy breach** might be just a matter of time, and nobody wants that
  - Sandboxing / App Store reviews are necessary, they should be kept and improved
  - **Risks must be known** and fairly evaluated

Recap

- You've seen iPhone main **privacy issues**
- You know which **personal data are at risk**
- You know how **spyware access these data**
- You've seen some potential **attack scenarios**
- I hope you will **use / deploy iPhones wisely**
- Contact me: nicolas@seriot.ch, Twitter @nst021
- **Time for Q&A**

Thank you!

Appendix:

Private APIs

Private APIs

- **Undocumented APIs**
- **Not allowed on the AppStore**
- SpyPhone does not use private APIs
  - Strings could be obfuscated or set remotely
  - Even **more data available** for spywares

```
NSString *path = @"/System/Library/PrivateFrameworks/Message.framework";
BOOL bundleLoaded = [[NSBundle bundleWithPath:path] load];

Class NetworkController = NSClassFromString(@"NetworkController");
NSString *IMEI = [[NetworkController sharedInstance] IMEI];
```

Appendix:
Swiss Law

Swiss Constitution

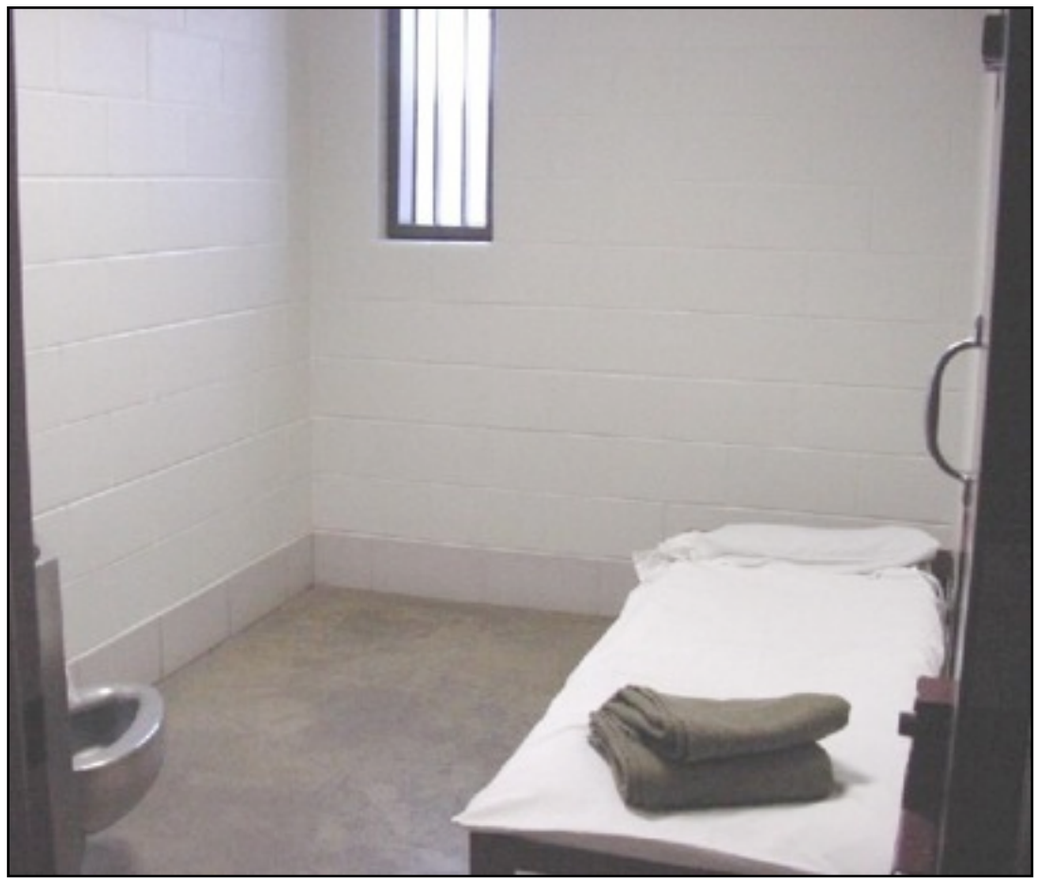
Protection of Privacy – Every person has the right to be protected against abuse of personal data (Art. 13 al. 2).

Personal Data

- **Personal data** : all information relating to an identified or identifiable person.
- **Personality profile** : permits an assessment of the essential characteristics of the personality of a natural person. Personality profiles are especially protected and strictly regulated.

Laws for Spyware Authors

- May be **jailed for up to three years**
- May have to **pay hefty fines**
- This is **scarcely applied** though



License Agreements

- **End users are protected** from over reaching End User License Agreements (EULAs).
- The EULA cannot simply state that you agree to send your personal data to bad guys if you do not.
- **There must be a real mutual agreement**, ruling out the use of potentially misleading terms.

Laws for Technical Staff

- In case of damages, **civil liability may apply to technical staff** if the plaintiff can prove that an organization failed to protect confidential data properly.

- **Liability could extend all the way to Apple itself.**