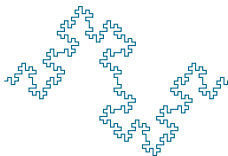


# Security Analysis of Android Bank Applications

Eric Filiol and Paul Irolla

*Laboratoire de Cryptologie et Virologie Opérationnelles  
École Supérieure d'Informatique, Électronique, Automatique*



31C3 - Hamburg - December 27<sup>th</sup>, 2014

# Table of contents

## Introduction & Background: the DAVFI Project

### Tools

- Static analysis

- APK Database

- Dynamic analysis

### Results on a Few Apps

- A Few Statistics

- JP Morgan Access

- BNP Paribas

- Sberbank

- Bradesco

### Conclusion & Future Work

# DAVFI Project

- ▶ Two-year project to develop a sovereign and trusted AV for Android, Linux and Windows.
- ▶ Funded partly by the French Government (6 millions euros with 0.35 % of funding).
- ▶ Intellectual property transferred to Nov'It for marketing under the brand *Uhuru.Mobile* and *Uhuru-AM*.
- ▶ Normally, free/open versions should be released by Nov'It for non commercial use.
- ▶ We will deliver a free/open fork version for Linux by mid March (OpenDAVFI Linux).
- ▶ More info on [http://www.davfi.fr/index\\_en.html](http://www.davfi.fr/index_en.html).

# DAVFI Android

- ▶ Transferred to Nov'It on October 17th, 2013.
- ▶ Based on Cyanogen and AOSP sources.
- ▶ We switched from a simple AV application to a complete antiviral operation system based on Android with additional security features:
  - ▶ File system encryption, SMS encryption, VoIP encryption, dedicated secure and certified application market...
- ▶ One of the key features is that all app available on the secure market is fully analyzed (static & dynamic analysis including possible reversing steps).
- ▶ Whenever safe AND compliant to our security policy (see further), the app is certified & signed before put on the secure market.
- ▶ More info on <http://www.uhuru-mobile.com/>

# Trust Policy

- ▶ Legit apps can be malevolent when it comes to targeted marketing and user tracking capabilities.
- ▶ A few apps contains severe vulnerabilities.
- ▶ The 'malware' definition needs to be extended.

## Trust Policy (Contnd)

An app is trustworthy according to our *Trust Policy* if and only if:

- ▶ It does not contain hidden functionalities.
- ▶ User informations collection must be motivated by explicit functionalities.
- ▶ Web communications involving personal user informations must be encrypted.
- ▶ The app does not contain known vulnerabilities.

# Why Bank Apps?

- ▶ Progressively, banks are forcing users to move towards mobile banking.
- ▶ Because our money is a serious business.
- ▶ Our privacy and data confidentiality is an even more critical issue!
- ▶ So, we expect them to be at the edge of security and confidentiality and to take care of our core interests.
- ▶ All banks have been contacted to provide (for free) all technical details. Up to now, only a very few have answered.
- ▶ A few (BNP Paribas, CA) are currently correcting part of the problems reported.

# Tools

- ▶ About 1800 malware from *malgenome* project and *contagiodump*.
- ▶ About 1800 genuine open sources gathered from *fdroid* and *Google code* projects.
- ▶ Tools we have developed:
  - ▶ *Egide*: advanced static analysis and malware detection tool.
  - ▶ *Tarentula*: web crawling tool to collect apps.
  - ▶ *Panoptes*: advanced dynamic analysis tool (network communications analysis at runtime).
- ▶ These tools are non public at the present time.



# Static analysis- Egide

- ▶ A program that reverses apps and generates a report which is a guide in the source code.
- ▶ Tasks: reverses to smali/java, detects risky behaviors/methods/sources/sinks, computes the control flow graph through entry point methods, computes statistics on group of apps, computes similarities between an app and a group of apps.
- ▶ Generates a neural network and trains it on an app database, generates reports and graphs...
- ▶ Demos and examples of reports.

# Static analysis- Egide - Exposed

```
trolla@porteurSain{ /Egide} - ./egide --noCache --scan ~/Téléchargements/bankApps/
Scanning 27 apks ...

com.BMCE_prod.bad.apk scanned
com.android.bankabc.apk scanned
```

| PID   | USER   | PRI | NI | VIRT | RES  | SHR  | S | CPU% | MEM% | TIME+   | Command  |
|-------|--------|-----|----|------|------|------|---|------|------|---------|--|
| 89988 | trolla | 20  | 0  | 264M | 140M | 5716 | R | 95.9 | 0.9  | 1:59.25 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 89923 | trolla | 20  | 0  | 225M | 101M | 5592 | R | 89.8 | 0.6  | 2:03.21 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 89997 | trolla | 20  | 0  | 227M | 103M | 5592 | R | 86.6 | 0.6  | 1:42.64 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 80030 | trolla | 20  | 0  | 233M | 109M | 5716 | R | 82.4 | 0.7  | 1:59.14 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 89991 | trolla | 20  | 0  | 228M | 103M | 5656 | R | 81.9 | 0.6  | 1:46.82 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 89994 | trolla | 20  | 0  | 246M | 122M | 5592 | R | 81.4 | 0.8  | 1:50.74 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |
| 89985 | trolla | 20  | 0  | 241M | 116M | 5704 | R | 76.3 | 0.7  | 1:36.60 | ruby ./egide --noCache --scan /home/trolla/Téléchargements/bankApps/ |

```
trolla@porteurSain{ /Egide} - evince scanReport/com.BMCE_prod.bad.apk/report.pdf
```

# Static analysis- Egide

- ▶ Proved its effectiveness on new malwares.
- ▶ But not entirely satisfying.
- ▶ A few samples never get well classified.
- ▶ The malware database needs to be extended.

# APK Database- Context

- ▶ Database of classified applications is the sinews of antiviral war.
- ▶ A subject rarely explained or detailed in security papers.
- ▶ A sophisticated data mining algorithm is useless with a poor database.
- ▶ So how to populate a database for training some edgy machine learning algorithm ?
- ▶ How does the others do?

# APK Database- Howto

- ▶ Several universities gather malware and propose to share them.
  - ▶ <http://www.malgenomeproject.org/>
  - ▶ <http://user.informatik.uni-goettingen.de/~darp/drebin/>
- ▶ Some websites share Android malware
  - ▶ <http://virusshare.com/>
  - ▶ <http://contagiodump.blogspot.fr/>
- ▶ It is a good starting point but not enough.

# APK Database- AV malware DB

From <https://www.virustotal.com/en/faq/>

## 🔧 Including new antivirus solutions and tools in VirusTotal

I would like to include my antivirus product/URL analysis engine in VirusTotal, what should I do?

The process could not be easier, just [contact us](#). We will tell you what we need.

In exchange for providing an antivirus solution you will receive all files submitted to VirusTotal that are not detected by your product and are detected by at least one other antivirus, along with their corresponding VirusTotal reports.

In exchange for allowing us to use a URL analysis engine you will receive the whole feed of URLs submitted to VirusTotal, along with their corresponding VirusTotal reports.

From [https://iseclab.org/papers/andrubis\\_badgers14.pdf](https://iseclab.org/papers/andrubis_badgers14.pdf)

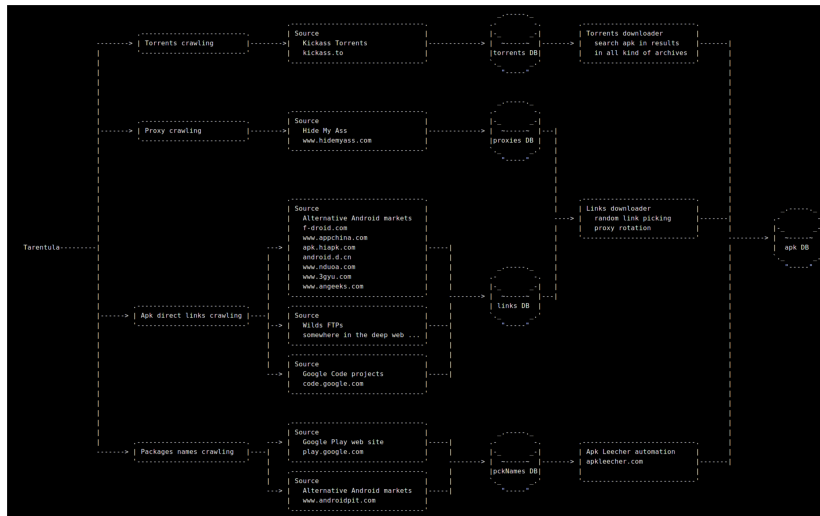
**TABLE II:** Sources for apps in our dataset: sample exchange feeds have a high proportion of malware, while the Google Play Store and apps from torrents and direct downloads have low infection rates. Interestingly, not all samples from malware corpora are detected by AV scanners.

| Category | Sample Exchange | Google Play | Alternative Markets | VirusTotal | Malware Corpora | Torrents | Direct Downloads | Unknown |
|----------|-----------------|-------------|---------------------|------------|-----------------|----------|------------------|---------|
| All      | 683,842         | 125,602     | 60,951              | 37,499     | 5,997           | 17,916   | 1,704            | 159,040 |
| Goodware | 5.2%            | 88.73%      | 18.15%              | 0.20%      | 0.04%           | 88.50%   | 96.36%           | 78.65%  |
| Malware  | 55.3%           | 1.60%       | 27.51%              | 98.65%     | 97.87%          | 1.60%    | 1.59%            | 7.56%   |

# APK Database- You Said Web Crawling?

- ▶ One tool at your disposal for free.
- ▶ Gather hundreds of thousands apk from questionable sources and hope to find dirty apps.
- ▶ Find sites, alternative markets, reverse them and fire the web crawler!

# APK Database- Tarentula Structure





# Dynamic analysis- Panoptes - Briefly

- ▶ Task: reveals communications between an app and internet.
- ▶ Open a fake access point and listen HTTP/HTTPS/POP/IMAP communications.
- ▶ Generates a tree of communication informations.

# Dynamic analysis- Panoptes - Does it Bypass SSL Encryption?

- ▶ A fake Certification Authority is in the phone.
- ▶ SSL/TLS requests are intercepted, terminated and a new one is initiated to the original destination address.
- ▶ The server response is copied, embedded in a SSL layer and signed with our fake Certification Authority.
- ▶ Panoptes demo.

# Apps Analyzed

At the present we have analyzed 27 apps (more to come in the forthcoming weeks).

BNP (France)

Crédit Agricole (France)

Société Générale (France)

Finaref (France)

BMCE (Morocco)

UBS (Switzerland)

Wells Fargo (USA)

Burke and Herbert (USA)

Commerzbank (Germany)

HSBC (UK)

Sberbank (Russia)

Agricultural Bank of China (China)

Hapoalim Bank (Israel)

HanaNBank (Korea)

LCL (France)

Sofinco (France)

BforBank (France)

Bradesco (Brazil)

Barclay (UK)

JP Morgan (USA)

Bank of America (USA)

PNC Financial Service (USA)

Deutsche Bank AG (Germany)

Santander Group (Spain)

Mizohobank (Japan)

State Bank of India (India)

Shahr Bank Iran

# A Few Statistics - Permissions

|                               |      |                                |    |
|-------------------------------|------|--------------------------------|----|
| INTERNET                      | 100% | SEND_SMS                       | 7% |
| ACCESS_NETWORK_STATE          | 96%  | RESTART_PACKAGES               | 7% |
| ACCESS_FINE_LOCATION          | 71%  | CHANGE_NETWORK_STATE           | 7% |
| WRITE_EXTERNAL_STORAGE        | 68%  | READ_SMS                       | 7% |
| READ_PHONE_STATE              | 61%  | RECORD_AUDIO                   | 7% |
| CAMERA                        | 54%  | READ_LOGS                      | 7% |
| ACCESS_COARSE_LOCATION        | 54%  | ACCESS_LOCATION_EXTRA_COMMANDS | 7% |
| c2dm.permission.RECEIVE       | 46%  | KILL_BACKGROUND_PROCESSES      | 7% |
| CALL_PHONE                    | 39%  | ACCESS_NETWORK                 | 4% |
| ACCESS_WIFI_STATE             | 39%  | GET_TASKS                      | 4% |
| READ_CONTACTS                 | 32%  | RECEIVE_MMS                    | 4% |
| gsf.permission.READ_GSERVICES | 29%  | MOUNT_UNMOUNT_FILESYSTEMS      | 4% |
| GET_ACCOUNTS                  | 29%  | DISABLE_KEYGUARD               | 4% |
| ACCESS_MOCK_LOCATION          | 14%  | READ_OWNER_DATA                | 4% |
| READ_EXTERNAL_STORAGE         | 14%  | READ_CALENDAR                  | 4% |
| RECEIVE_BOOT_COMPLETED        | 14%  | WRITE_CALENDAR                 | 4% |
| WRITE_CONTACTS                | 11%  | BROADCAST_STICKY               | 4% |
| NFC                           | 11%  | SMARTCARD                      | 4% |
| RECEIVE_SMS                   | 11%  | NFC_TRANSACTION                | 4% |
| WRITE_SETTINGS                | 11%  | ACCESS_DOWNLOAD_MANAGER        | 4% |
| CHANGE_WIFI_STATE             | 11%  | READ_CALL_LOG                  | 4% |

# Some statistics - Behaviors

|                                   |     |
|-----------------------------------|-----|
| Load app content from web         | 96% |
| Can use clear text communications | 89% |
| Get OS name                       | 75% |
| Get android unique id             | 71% |
| Use addJavascriptInterface        | 61% |
| Get IMEI                          | 54% |
| Get OS version                    | 54% |
| User tracking capabilities        | 50% |
| Get MAC address                   | 25% |
| Get MSISDN                        | 18% |
| Get IMSI                          | 11% |
| Get CID                           | 7%  |
| Get LAC                           | 4%  |
| Get phone serial number           | 4%  |
| Get router MAC address            | 4%  |



# JP Morgan Access

“oxrohccRtI/m1w9NC/7nqwANljaa8fORRXcJ2S1EiThNdeuW6GER  
L7NQogAnOFtPdYlwP1Gh2+0aNqsnrKeGbw==

#####

MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAMx6N9b4yaIFC60o  
f8YWU1e08sh4KRoldfjRKmtVazOKg2p3UUwMT5oUwBYYEhWsSl  
+bTD6DMCIQrwr2iSW09DkCAwEAAQ==

#####

Root Call Blocker,LBE Privacy Guard,Dual Mount SD Widget,  
Hexamob Recovery Pro,Total Commander,Boot Man”

# JP Morgan Access

```
String[] arrayOfString = str2.split("#####");
```

```
...
```

```
RootTools.log("Executing sh Commands : " +  
arrayOfString5[0] + arrayOfString5[1]);
```

```
...
```

```
List localList = runcmd(arrayOfString6);
```



# JP Morgan Access

- ▶ Dynamic Analysis :
  - ▶ An encrypted string is received.
  - ▶ APK Instrumentation reveals it contains signatures and lists of strings.
- ▶ Static Analysis reveals some strings are sent directly into a shell.
- ▶ Well, that's a remote shell, isn't it?

# BNP Paribas

Demo

# BNP Paribas

- ▶ Dynamic analysis failed!
- ▶ Static analysis reveals that *addJavascriptInterface* loads plaintext javascript.
- ▶ That means MITM attackers can gain a reverse shell on vulnerable phones.

# Sberbank

Demo

# Sberbank

- ▶ wifinetworks=001122334400:-45,0060B3E268C8:-66,4018B1CF2655:-77,4018B1CF2255:-77,4018B1CF6455:-79,C8D3A352B1B0:-78,4018B1CF6515:-83,586D8F747EC7:-85,4018B1CF2654:-76,4018B1CF2254:-83,4018B1CF6454:-84,4018B1CF6514:-88,4018B1CF23D4:-90,4018B1CF23D5:-83,4018B1CF63D4:-92,D8C7C8138A92:-90
- ▶ 001122334400 it is MAC address of my wifi access point used for interception.
- ▶ So the app send surrounding wifi networks MAC addresses and signal strength, in plaintext.

# Sberbank

- ▶ Dynamic analysis reveals all surroundings wifi networks info are sent in plaintext to Yandex servers.
- ▶ Static analysis reveals that it is used for fine indoor geolocation.
- ▶ In fact, Google maps services does it too which is installed on every Android phone.
- ▶ That's basically world wifi networks mapping.
- ▶ "Hello Google, someone has stolen my wifi router, can you send me its coordinates please?"

# Bradesco

Demo

# Bradesco

- ▶ Dynamic analysis reveals a private key for accessing bank services is received in plaintext.
- ▶ The embedded *Jquery javascript* lib contains vulnerabilities.



# Future work

- ▶ We intend to cover all banking apps throughout the world.
- ▶ Other kind of apps will be analyzed (games, email clients, security tools...).
- ▶ Develop out tools further with advanced mathematics (Ph D starting in 2015).
- ▶ Publish the {Egide, Panoptes} reports once security issues will be corrected.
- ▶ Verification analysis will be performed to check whether the users' privacy issues have been solved as well.

# Conclusion

- ▶ Banking apps are far from being totally clean. Beyond a few cases of vulnerabilities, users' privacy is not the priority of banks!
- ▶ There is a strong need for pressure on app developers to take care of users' privacy.
- ▶ The bank apps market is not mature and has developed too quickly. Functionalities take precedence over security and users' fundamental rights for privacy and data confidentiality.
- ▶ It is very difficult to identify a visible contact point to report security issues.
- ▶ All the tested apps are on the *Google Play*!
  - ▶ This means that Google does not perform apps' security analysis at all! It does not care about users' privacy either (but we all already know that).

Many thanks for your attention

Questions & Answers

Contact:

`filiol@esiea.fr`