(In)Security of Mobile Banking

Eric Filiol and Paul Irolla
Laboratoire de Cryptologie et Virologie Opérationnelles
École Supérieure d'Informatique, Électronique, Automatique

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

31C3 - Hamburg - December 27^{th}, 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`.

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.

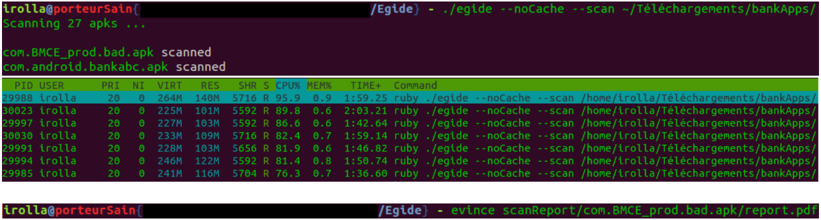
Tools

- About 1800 malware from *malgenome* project and *contagiodump*.
- About 1800 genuine open sources gathered from *fdroid* and *Google code* projects.
- Tools we have developped:
  - *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 similarites between an app and a group of apps.
- Generates a neural network and trains it on an app database, generates reports and graphes...
- Demos and examples of reports.

Static analysis- Egide - Exposed



```
irolla@porteurSain{[redacted]/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
29988 irolla     20   0  264M  140M  5716 R 95.9  0.9  1:59.25 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
30023 irolla     20   0  225M  101M  5592 R 89.8  0.6  2:03.21 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
29997 irolla     20   0  227M  103M  5592 R 86.6  0.6  1:42.64 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
30030 irolla     20   0  233M  109M  5716 R 82.4  0.7  1:59.14 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
29991 irolla     20   0  228M  103M  5656 R 81.9  0.6  1:46.82 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
29994 irolla     20   0  246M  122M  5592 R 81.4  0.8  1:50.74 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
29985 irolla     20   0  241M  116M  5704 R 76.3  0.7  1:36.60 ruby ./egide --noCache --scan /home/irolla/Téléchargements/bankApps/
```

```
irolla@porteurSain{[redacted]/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

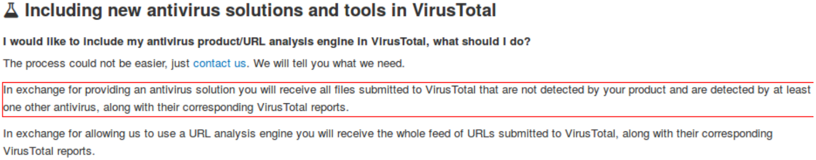
- Database of classified applications is the sinews of antiviral war.
- A subject rarely explained or detailed in security papers.
- A sophisticated data mining algorithm is useless with a poor database.
- So how to populate a database for training some edgy machine learning algorithm ?
- How does the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

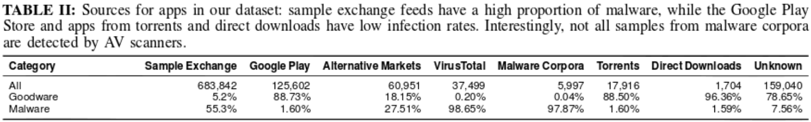


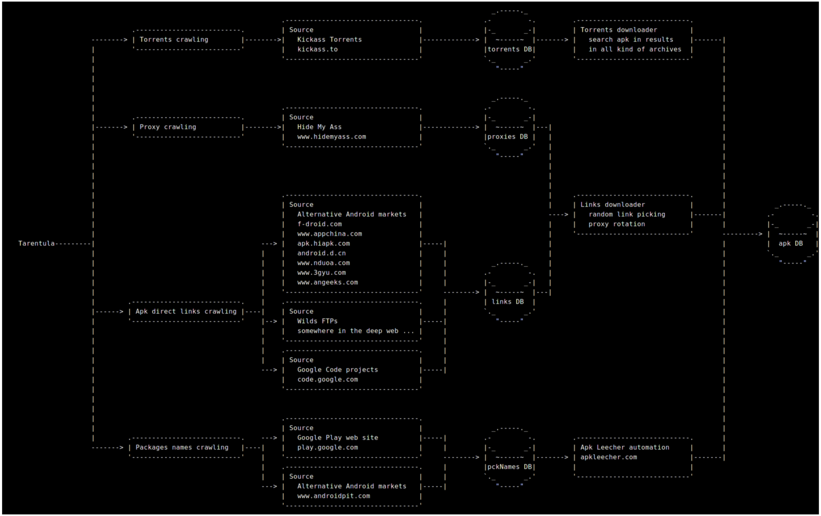
TABLE II: Sources for apps in our dataset: sample exchange feeds have a high proportion of malware, while the Google Play Store and apps from torrents and direct downloads have low infection rates. Interestingly, not all samples from malware corpora are detected by AV scanners.

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- Tarentula

- Crawl alternative Android market from USA, China, Russia etc.
- Crawl public ftp.
- Crawl torrents.
- Crawl Google Play.
- We have stopped crawling at 280K unique apps.

APK Database- Tarentula Structure



Tarentula

Torrents crawling

Source
Kickass Torrents
kickass.to

torrents DB

Torrents downloader
search apk in results
in all kind of archives

Proxy crawling

Source
Hide My Ass
www.hidemyass.com

proxies DB

Source
Alternative Android markets
f-droid.com
www.appchina.com
apk.hiapk.com
android.d.cn
www.nduoa.com
www.3gyu.com
www.angeeks.com

Links downloader
random link picking
proxy rotation

apk DB

links DB

Apk direct links crawling

Source
Wilds FTPs
somewhere in the deep web ...

Source
Google Code projects
code.google.com

Packages names crawling

Source
Google Play web site
play.google.com

pckNames DB

Apk Leecher automation
apkleecher.com

Source
Alternative Android markets
www.androidpit.com

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

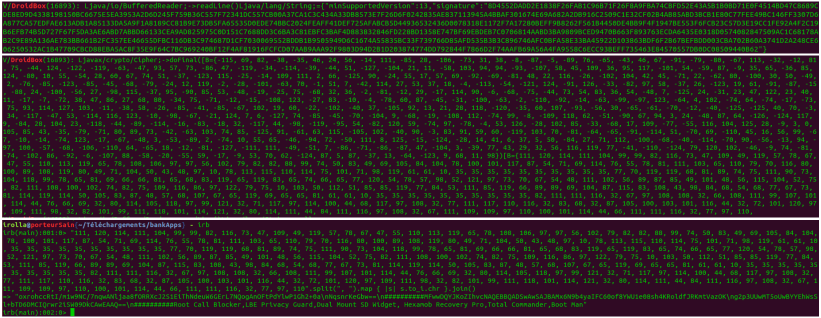
INTERNET	100%	SEND_SMS	7%
ACCESS_NETWORK_STATE	96%	RESTART_PACKAGES	7%
ACCESS_FINE_LOCATION	71%	CHANGE_NETWORK_STATE	7%
WRITE_EXTERNAL_STORAGE	68%	READ_SMS	7%
READ_PHONE_STATE	61%	RECORD_AUDIO	7%
CAMERA	54%	READ_LOGS	7%
ACCESS_COARSE_LOCATION	54%	ACCESS_LOCATION_EXTRA_COMMANDS	7%
c2dm.permission.RECEIVE	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

Demo



```
V/DroidBox(16893): Ljava/io/BufferedReader;->readLine()Ljava/lang/String;={"minSupportedVersion":13,"signature":"8D4552DADD2E1838F26FAB1C96B71F26F8A9FBA74CBFD52E43A5B1B0BD71E0F4514BD47C8689C
DE8ED9D4338198150BC6675E5EA3953A2D6D245F759B36C557F72341DC557CB00A37CA1C3C434A33DB8573E7F26D6F0242835AE837113945A4BBAF301674E69A682A2DB916C2509C1E32CF02B4AB85ABD3BCB1E80C77FEE49BC146FF3307D6
A877CA57EDFAE613ADB1A85133DA5A9F1A8189CC81B9E73DB5FA65535D0EDE74BBC2024FEAFF41DEF725AFABCB5D44936532436D0078318E1172F7A17280BEFF9B8262F561B4450DE4BB9F4F1947BE553F6FCB23C57D3E19CC1FE92A4F2C19
B6EFB74B5D727F67F5DA3AE6ABD7ABBD66133CEA9AD825975C0D151C7688DD3C6BA3C81EBFC3BAF4D883832846FD228BD1358E747BF69EBDEB7C0706814AABD3BA9809BCED9470B663F893763ECDA6435E0318D0574082847509AC1C68178A
B2C9E89A136AE783BB661B2FC357EE46655DFBC116DB3C974687D1CF7030069552BDDB1B9505949D6C1674A55835BC33F739766D85AFD535B3B3C896746AFC0BFA58E33BA45922D103863BDF6F2B67BEFBDD003CBA702B60A3741D2A248CE6
06250532AC1B47709CBCD88EBA5AC8F35E9F64C7BC969240BF12F4AF81916FCFCD07AAB9AAA92F9803D94D2B1D203874774DD792844F7866D2F74AAFB69A56A4FA9558C6ECC93BEFF735463E84570557DB0DC08509440B62"}
```

```
V/DroidBox(16893): Ljavax/crypto/Cipher;->doFinal([B={-115, 69, 82, -38, -35, 46, 24, 56, -14, 111, -85, 28, -106, -73, 31, 38, -8, -87, -5, -89, 76, -65, -43, 46, 67, -91, -79, -80, -67, 113, -32, -12, 81
, 75, -44, 124, -122, -119, -63, -47, 93, 57, 73, -86, 47, -119, -34, -114, -39, -44, 51, -127, -104, 21, 11, -58, 103, 94, 94, -93, -107, 58, 45, 109, 36, 95, 117, -101, 54, -59, 87, -9, 35, 65, -36, 85,
124, -80, 10, 55, -54, 28, 60, 67, 74, 51, -37, -123, 115, -25, -14, 109, 111, 2, 66, -125, 90, -24, 55, 17, 57, 69, -92, -69, -81, 48, 22, 116, -26, -102, 104, 42, 45, -71, 22, -62, 80, -100, 30, 50, -49,
 2, -76, -85, -123, -85, -45, -68, -79, -24, 12, 119, -2, -28, -101, -63, 70, -1, 51, 7, -42, 114, 27, 53, 37, 18, -4, -113, -54, -121, 124, -91, 126, -33, -82, 97, 58, -37, 26, -123, 19, 61, -91, -87, -15
, -88, 24, -100, -56, 27, -98, 115, -37, 95, -90, 85, 53, -48, -19, -25, 75, -68, 32, 36, -2, -81, -12, 29, -17, 114, 90, -6, -68, -75, -44, 73, 54, 83, 36, 54, -48, 7, -125, 24, -31, 23, 47, 122, 23, 40,
11, -17, -7, -72, 38, 47, 86, 27, 68, 80, -34, 75, -71, -12, -15, -108, 123, -27, 83, -10, -4, -78, 60, 87, -45, -31, -100, -63, -2, -110, -92, -14, -63, -99, -97, 123, -64, 4, 102, -74, 64, -74, -17, -73,
 75, 93, 114, 127, 103, -11, -38, 58, -26, -85, -41, -85, -67, 102, 19, 60, -22, -102, -40, 37, -105, 92, 13, 21, 28, 118, -120, -35, 60, 107, -93, -56, 30, -65, -61, -70, -12, -40, -125, -125, 40, 70, -3,
 34, -117, -47, 53, -114, 116, 123, -10, -98, -67, -21, 124, 7, 6, -127, 74, -85, -45, -70, -104, 9, -68, -19, -108, 112, -74, 99, -8, -109, 118, 62, -51, -90, 67, 94, 3, 24, -48, 87, 64, -126, -124, 117,
9, -84, 28, 104, 23, -118, -44, -89, -114, -16, -83, -18, 32, -117, 44, -98, -119, -95, 54, -82, 120, 59, -74, 97, -78, -4, 53, 126, -28, 102, 85, -33, -68, 17, 109, -77, -55, 116, 104, 125, 28, -9, 3, 0,
105, 85, 43, -35, -79, -71, 80, 89, 73, -42, -63, 103, 74, 85, -125, 91, -61, 63, 115, -105, 102, -40, 90, -3, 83, 91, 59, 60, -119, 103, 70, -81, -64, -65, -91, -114, 51, -70, 69, -110, 45, 16, 56, 99, -6
7, -10, -14, -74, 123, -17, -67, -48, 3, -53, -89, 2, -74, 10, 55, 65, -46, -94, 72, -50, 111, 8, 125, -15, -124, -28, 14, 41, 6, 37, 5, 50, -84, 27, 71, 112, -100, -68, -40, -114, -70, 90, -56, -13, 94, -
97, 100, -57, -68, -106, -110, 64, -65, 18, -12, -81, -127, -111, 111, -49, -51, 7, -86, -71, -86, -87, 47, -104, 3, -39, 77, 43, 29, 32, 56, 116, 119, 77, -41, -110, -124, 79, 120, 102, -46, -9, 74, -81,
-74, -102, 86, -92, -6, -107, 88, -58, -20, -55, 59, -17, -9, 53, 70, 62, -124, 87, 5, 87, -37, 13, -64, -123, 9, 68, 11, 98})[B={111, 120, 114, 111, 104, 99, 99, 82, 116, 73, 47, 109, 49, 119, 57, 78, 67,
 47, 55, 110, 113, 119, 65, 78, 108, 106, 97, 97, 56, 102, 79, 82, 82, 88, 99, 74, 50, 83, 49, 69, 105, 84, 104, 78, 100, 101, 117, 87, 54, 71, 69, 114, 76, 55, 78, 81, 111, 103, 65, 110, 79, 70, 116, 80,
100, 89, 108, 119, 80, 49, 71, 104, 50, 43, 48, 97, 10, 78, 113, 115, 110, 114, 75, 101, 71, 98, 119, 61, 61, 10, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 77, 70, 119, 119, 68, 81, 89, 74, 75, 111, 90, 73,
104, 118, 99, 78, 65, 81, 69, 66, 66, 81, 65, 68, 83, 119, 65, 119, 83, 65, 74, 66, 65, 77, 120, 54, 78, 57, 98, 52, 121, 97, 73, 70, 67, 54, 48, 111, 102, 56, 89, 87, 85, 49, 101, 48, 56, 115, 104, 52, 75
, 82, 111, 108, 100, 102, 74, 82, 75, 109, 116, 86, 97, 122, 79, 75, 10, 103, 50, 112, 51, 85, 85, 119, 77, 84, 53, 111, 85, 119, 66, 89, 89, 69, 104, 87, 115, 83, 108, 43, 98, 84, 68, 54, 68, 77, 67, 73,
81, 114, 119, 114, 50, 105, 83, 87, 48, 57, 68, 107, 67, 65, 119, 69, 65, 65, 81, 61, 61, 10, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 82, 111, 111, 116, 32, 67, 97, 108, 108, 32, 66, 108, 111, 99, 107, 101
, 114, 44, 76, 66, 69, 32, 80, 114, 105, 118, 97, 99, 121, 32, 71, 117, 97, 114, 100, 44, 68, 117, 97, 108, 32, 77, 111, 117, 110, 116, 32, 83, 68, 32, 87, 105, 100, 103, 101, 116, 44, 32, 72, 101, 120, 97
, 109, 111, 98, 32, 82, 101, 99, 111, 118, 101, 114, 121, 32, 80, 114, 111, 44, 84, 111, 116, 97, 108, 32, 67, 111, 109, 109, 97, 110, 100, 101, 114, 44, 66, 111, 111, 116, 32, 77, 97, 110,
```

```
irolla@porteurSain{~/Téléchargements/bankApps} - irb
irb(main):001:0> "111, 120, 114, 111, 104, 99, 99, 82, 116, 73, 47, 109, 49, 119, 57, 78, 67, 47, 55, 110, 113, 119, 65, 78, 108, 106, 97, 97, 56, 102, 79, 82, 82, 88, 99, 74, 50, 83, 49, 69, 105, 84, 104,
 78, 100, 101, 117, 87, 54, 71, 69, 114, 76, 55, 78, 81, 111, 103, 65, 110, 79, 70, 116, 80, 100, 89, 108, 119, 80, 49, 71, 104, 50, 43, 48, 97, 10, 78, 113, 115, 110, 114, 75, 101, 71, 98, 119, 61, 61, 10
, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 77, 70, 119, 119, 68, 81, 89, 74, 75, 111, 90, 73, 104, 118, 99, 78, 65, 81, 69, 66, 66, 81, 65, 68, 83, 119, 65, 119, 83, 65, 74, 66, 65, 77, 120, 54, 78, 57, 98,
 52, 121, 97, 73, 70, 67, 54, 48, 111, 102, 56, 89, 87, 85, 49, 101, 48, 56, 115, 104, 52, 75, 82, 111, 108, 100, 102, 74, 82, 75, 109, 116, 86, 97, 122, 79, 75, 10, 103, 50, 112, 51, 85, 85, 119, 77, 84,
53, 111, 85, 119, 66, 89, 89, 69, 104, 87, 115, 83, 108, 43, 98, 84, 68, 54, 68, 77, 67, 73, 81, 114, 119, 114, 50, 105, 83, 87, 48, 57, 68, 107, 67, 65, 119, 69, 65, 65, 81, 61, 61, 10, 35, 35, 35, 35, 35
, 35, 35, 35, 35, 35, 82, 111, 111, 116, 32, 67, 97, 108, 108, 32, 66, 108, 111, 99, 107, 101, 114, 44, 76, 66, 69, 32, 80, 114, 105, 118, 97, 99, 121, 32, 71, 117, 97, 114, 100, 44, 68, 117, 97, 108, 32,
77, 111, 117, 110, 116, 32, 83, 68, 32, 87, 105, 100, 103, 101, 116, 44, 32, 72, 101, 120, 97, 109, 111, 98, 32, 82, 101, 99, 111, 118, 101, 114, 121, 32, 80, 114, 111, 44, 84, 111, 116, 97, 108, 32, 67, 1
11, 109, 109, 97, 110, 100, 101, 114, 44, 66, 111, 111, 116, 32, 77, 97, 110".split(", ").map { |s| s.to_i.chr }.join()
=> "oxrohccRtI/m1w9NC/7nqwANljaa8fORRXcJ2S1EiThNdeuW6GErL7NQogAnOFtPdYlwP1Gh2+0a\nNqsnrKeGbw==\n##########MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAMx6N9b4yaIFC60of8YWU1e08sh4KRoldfJRKmtVazOK\ng2p3UUwMT5oUwBYYEhWsS
l+bTD6DMCIQrwr2iSW09DkCAwEAAQ==\n##########Root Call Blocker,LBE Privacy Guard,Dual Mount SD Widget, Hexamob Recovery Pro,Total Commander,Boot Man"
irb(main):002:0>
```

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:-78,4018B1CF6 515:-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 surrondings 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 developpers to take care of users' privacy.
- The bank apps market is not mature and has developped 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.

Conclusion

- 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).
  - Google has the power to force developpers to do a better job.
- Choose open source apps (when available, for banks, well it is utopia).
- Prefer local/national banks instead of international banks.

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

Many thanks for your attention

Questions & Answers

Contact: `filiol@esiea.fr`