Android Anti-Forensics Through a Local Paradigm

By

Alessandro Distefano, Gianluigi Me and Francesco Pace

Presented At

The Digital Forensic Research Conference

DFRWS 2010 USA Portland, OR (Aug 2nd - 4th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org

University of Roma
TOR VERGATA
Italy

Android Anti-Forensics Through a Local Paradigm

Gianluigi ME

INTRODUCTION

- Mobile phones, are among the most common and diffused current technologies:
  - 2.6 billion of subscribers in the world

- Class of Mobile Phones (NIST):
  1. Basic.
  2. Advanced.
  3. Smart.

- Regarding the forensic environment, a very large amount of personal information is stored into advanced/smartphones

STATE OF THE ART

- Mobile Forensics is still experiencing a number of difficulties and problems (mainly due to models ethereogeneity jungle and to the unremovable internal memory).

- Anti-Forensics (AF)
  - *"Any attempts to compromise the availability or usefulness of evidence in the forensic process"* (R. Harris – 2004)

- By the study of the AF techniques, a number of useful conclusions and guidelines can be drawn, in order to improve and harden the currently used forensic tools and techniques

KINDS OF ANTI-FORENSICS

1. Destroying Evidence
   - It involves the destruction of evidence, in order to make it unusable during the investigative process.
2. Hiding Evidence
   - It is the act of administrate the evidence in order to decrease, or even nullify, its visibility during the forensics analysis.
3. Eliminating Evidence Sources
   - It is the neutralization of the evidentiary sources.
4. Counterfeiting Evidence
   - It is the creation of a fake version of the evidence (Poisoning).

MOBILE ANTI-FORENSICS

- Classical forensic guidelines and tools, often, are not suitable for Mobile Devices as well.

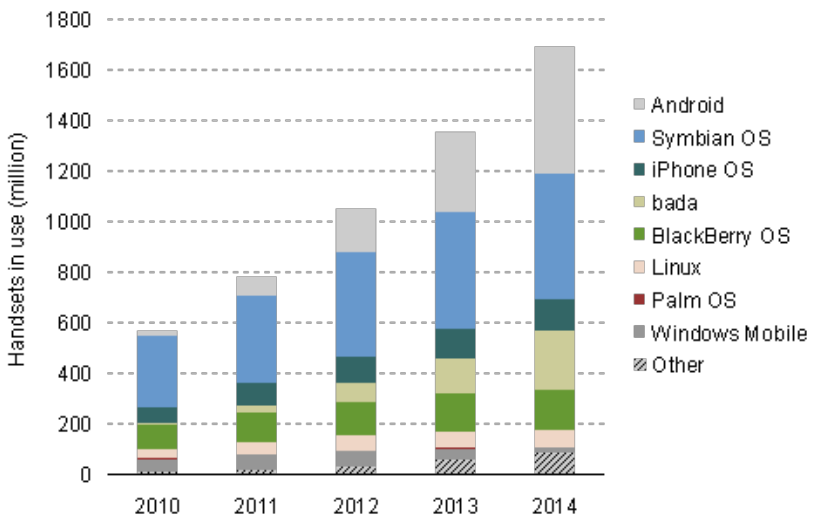
- **Problem**: unavailability of a direct access to the internal memory:
  - In fact, if the removable storage volumes (e.g., memory cards, SIM cards) can be isolated from the device and analyzed with standard procedures, the internal memory volume cannot.
  - The internal memory seems to be an ideal candidate in order to apply some AF techniques.

- However, as for any other commercial forensic tool, concerns on the tool behavior arise

ANDROID OS

- Android is a set of open source software elements specifically designed for Mobile Devices, it includes:
  1. Operating System (OS).
  2. Middleware.
  3. Set of native application.

Smartphone handsets in use, by OS, 2010–2014 [Source: Analysys Mason, 2010]

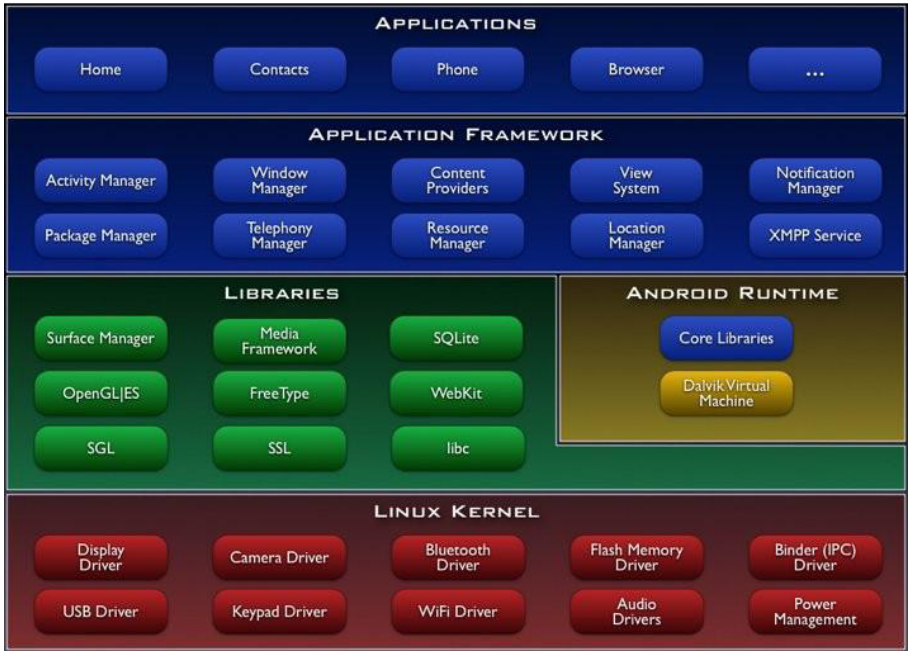


Top Smartphone Platforms 3 Month Avg. Ending May 2010 vs. 3 Month Avg. Ending Feb. 2010 Total U.S. Age 13+ Source: comScore MobiLens			
	Share (%) of Smartphone Subscribers		
	Feb-10	May-10	Point Change
Total Smartphone Subscribers	*100.0%*	*100.0%*	*N/A*
RIM	42.1%	41.7%	-0.4
Apple*	25.4%	24.4%	-1.0
Microsoft	15.1%	13.2%	-1.9
Google	9.0%	13.0%	4.0
Palm	5.4%	4.8%	-0.6

• Analysis Mason Forecasts confirms that the 2014 market share taken by Android will be approximately of 1.7 billion devices

ANDROID OS: OVERVIEW

- Android Architecture is composed by five major components:
  1. Applications.
  2. Application Framework.
  3. Libraries.
  4. Android Runtime.
  5. Linux Kernel.



- Android File System:
  - Natively supported YAFFS2.
  - Designed for NAND Flash chips.

ANDROID SECURITY ARCHITECTURE

- Multi-process platform which relies on the standard Linux facilities:
  - Security between applications is enforced at process level.
- Application & Sandboxes:
  - Android denies to any application the capability to perform operations with the objective to hamper any other application, the OS or the end-user.
- User Ids & Permissions:
  - Android manages every installed application as a different Linux user.
  - The applications have to export their service to the Manifest files,
    It's the only way to guarantee the communication between us.

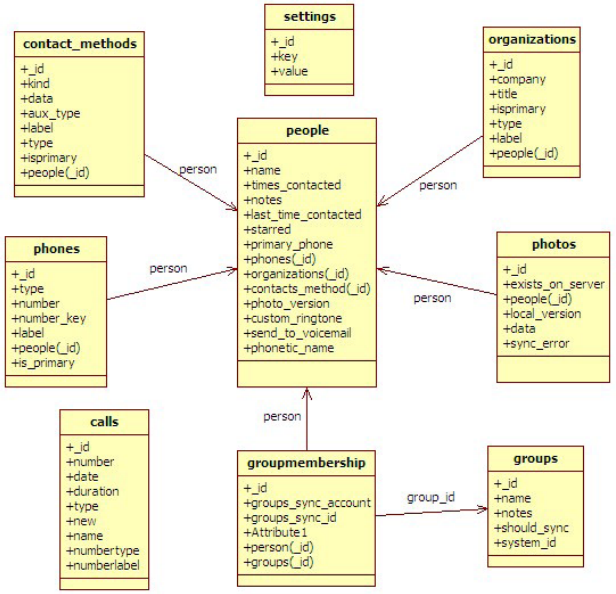
ANDROID ANTI-FORENSICS

- Three main concepts behind the work:
  1. Exploting Android Features.
  2. A Private Folder.
  3. Anti-Forensics by a Common Application.

- Thanks to the standard Android security features, for a given application it is possible to create a directory that is inaccessible for any other applications:
  - It used to store any kind of information (e.g., text files, multimedia).
  - It's created at install time and remove when the owning application is uninstalled.
  - Easy to figure out how this kind of folders can be exploited in order to perform AF Techniques.
  - Inaccessibility ensure the protection of the stored data.

ANDROID DATABASES

- Android OS store any kind of informations in sqlite3 databases within own application private folder, for example:
  - Contacts are in */data/data/com.android.providers.contacts/databases/contacts.db*
  - *SMS/MMS are in /data/data/com.android.providers.telephony/databases/mmssms.db*
  - *Media Files are in /data/data/com.android.providers.media/databases/external.db*



CONTACT DATABASE

• Any sqlite3 file has a restriction access, to provide application ownership these data;

• To read/write data into databases the applications must specify correct permission in **AndroidManifest.xml**, for example:

- *android.permission.READ_SMS*
- *android.permission.WRITE_SMS*

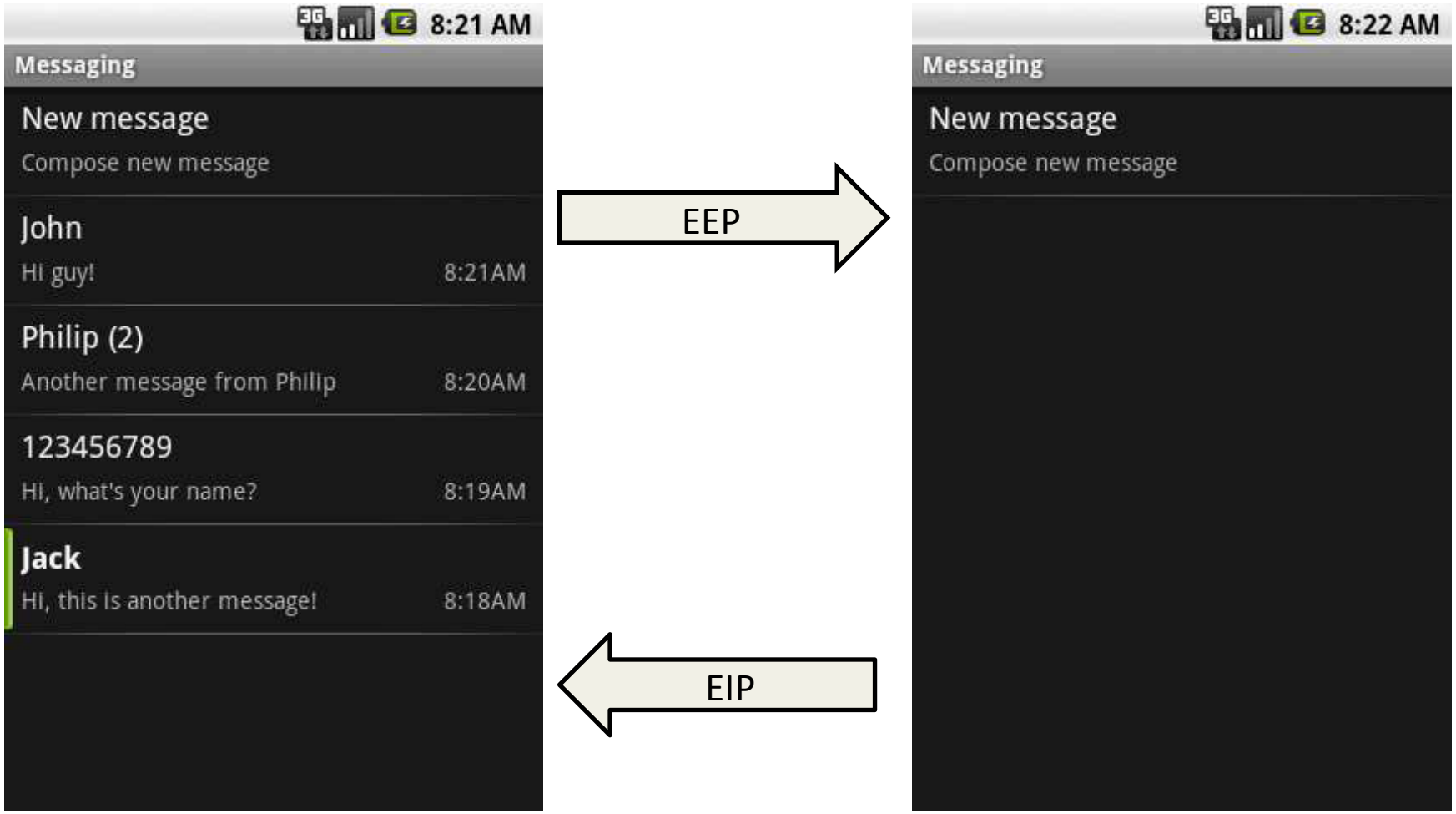
•Android AF analyzes the overall databases structure and execute some Update/Delete queries to apply AF Techniques

ANDROID ANTI-FORENSICS

- Private Folder features:
  - Data will be discovered only if the volume could be isolated.
  - Currently, isolation techniques and physical imaging are hard tasks.
  - It's impedes the cursory examination because infos are invisible to end-user.

- Implement AF as Android Application (AFDroid)
  - At install time, AFDroid creates private folder and it allows execution of two distinct processes:
    1. Evidence Export Process (EEP)
    2. Evidence Import Process (EIP)

ANDROID AF: EEP/EIP PROCESS



EEP: GOAL & FOCUS

- **Goal**: use the AF approach to delete/counterfeit evidence.

- For each technique developed the related feature exploiting the Android Application Framework:
  - *SMS/Call Logs* vs. *Destroying Evidence.*
  - *Contact* vs. *Counterfeiting Evidence.*
  - *Media Files* vs. *Hiding Evidence.*
  - *MMS* vs. *Eliminating Evidence Sources.*

EEP Idea: Producing an `export.xml` Containing the evidence gathered by the target Android databases. It's stored by private directory.

AF-TECHNIQUES ON EEP

- *Android Destroying Evidence:* deleting from the related databases any records which can carry sensitive information:
  - The investigator cannot find any information.
- *Android Hiding Evidence:* moving sensitive media files into the private folder:
  - The multimedia management applications cannot index the data.
- *Android Eliminating Evidence Sources:* it's enough to tamper the mechanism of conversation identifiers:
  - Any related MMS cannot be properly indexed by the system.
- *Android Counterfeiting Evidence:* *it's enough to change* a flag that states if the contact is among the preferred ones, and the related number of performed interactions:
  - This evidence can lead to a fast identification of strong relations between contacts.

EIP: GOAL & FOCUS

- **Goal**: restore the last state evidence stored inside the device before the EEP process

- Fully automated evidence reconstruction:
  - By Private Folder inspection.
  - XML File processing by SAX-XML Parser.
  - Other file processing.

- Evidence reconstruction and Forensic properties:
  - Automatic process for the reconstruction leverages on the capability of restoring both the generic files and databases contents.
  - EIP is reversible from the perspective of the end-user.

EIP: EXAMPLE

- Capability to restore the previous state of the device reading `export.xml` file

SMS Example

```
<database name='MMSSMS'>
<table name='sms'>
<row>
    <col name='_id'>977</col>
    <col name='thread_id'>15</col>
    <col name='address'>YYYYYYYYYYYY</col>
    <col name='person'>1148</col>
    <col name='date'>1265591133661</col>
    <col name='protocol'>0</col>
    <col name='read'>1</col>
    <col name='status'>-1</col>
    <col name='type'>1</col>
    <col name='reply_path_present'>0</col>
    <col name='subject'>null</col>
    <col name='body'>Text of the message</col>
    <col name='service_center'>XXXXXXXXXXXXX</col>
</row>
</table>
</database>
```

CONTACT Example

```
<database name='CONTACTS'>
<table name='calls'>
<row>
    <col name='_id'>2896</col>
    <col name='number'>YYYYYYYYYYYY</col>
    <col name='date'>1263580272900</col>
    <col name='duration'>288</col>
    <col name='type'>1</col>
    <col name='new'>1</col>
    <col name='name'>NameOfContact</col>
    <col name='numbertype'>2</col>
    <col name='numberlabel'>null</col>
</row>
</table>
</database>
```

EXPERIMENTS

- *Objectives:* test the strength of the selected processes in relation to the tools that are currently able to acquire a snapshot of the internal memory of the target device:
  - the strength of a given process that instantiates some AF techniques is inversely related to the capability to recover the processed evidence.

- *Used Devices:* experiments were performed on most recently smartphone:
  - Samsung Galaxy i7500, 1.6 SDK (*Kernel* 2.6.29, *Build* Donut.XEJC6)
  - HTC Magic 32b, 2.1-update1 SDK (*Kernel* 2.6.34, *Build* EPE54B)

- *Used Acquisition Tools:*
  - MIAT for Android (http://www.miaforensics.org)
  - Nandroid

EXPERIMENTS

- *Experimental Workflows:* formed by two main process
  - Evidence Export Process – EEP
  - Evidence Destruction Process – EDP

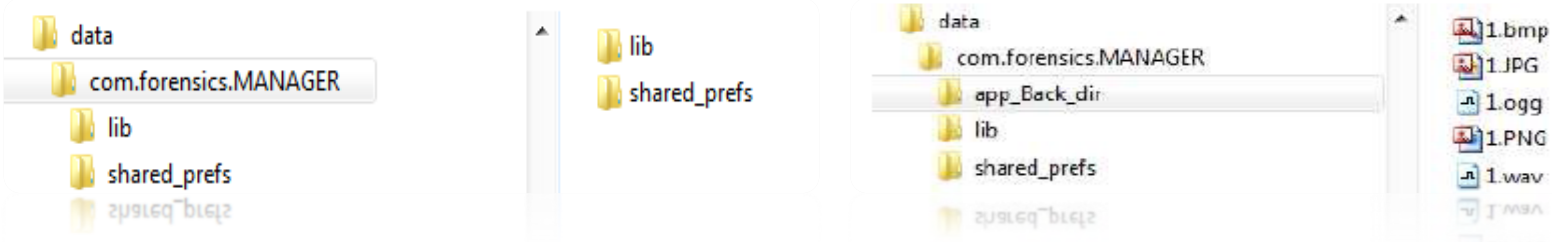
- *Experimental Results:* considered two different kinds of analysis of the target device:
  - *Cursory examination.*
  - *Acquisition & Analysis of the internal memory.*

File	Before	After
com.android.browser/databases/browser.db	58	5
com.android.providers.telephony/databases/mmssms.db	189	48

SIZE DIFFERENCES (KB) BETWEEN THAT FILES THAT STORE THE DATABASE AFFECTED BY THE EEP

EEP – EXPERIMENTS ANALYSIS

- After this task, any cursor examination of the device shows the following situation:
  - *Contacts:* no differences in terms of number of interactions.
  - *SMS/MMS/Call Log*: databases is empty.
  - *Multimedia Gallery*: empty folders.



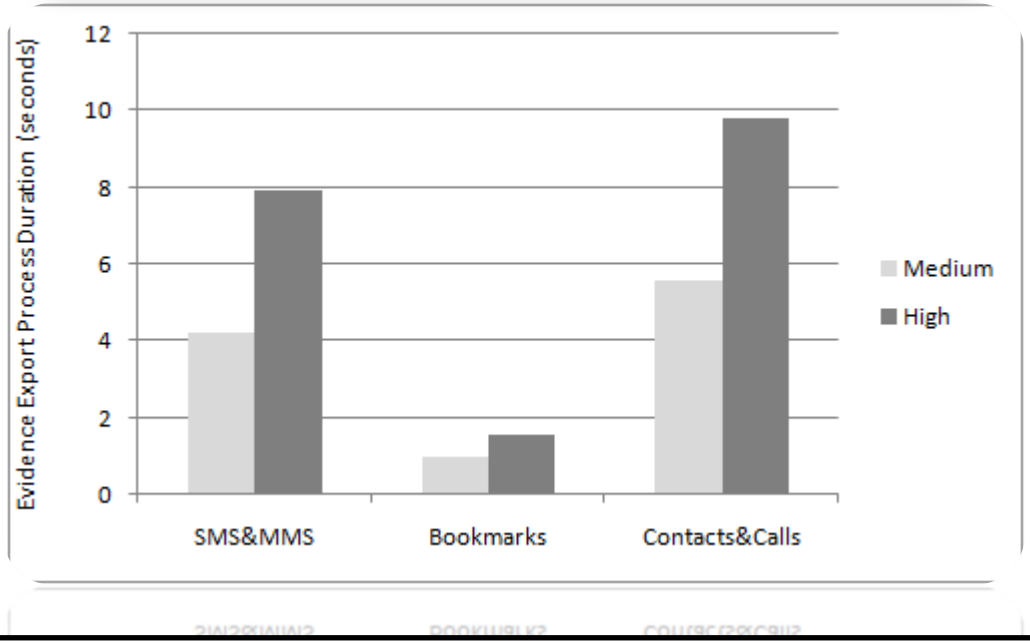
BEFORE EEP AFTER EEP

- NANDROID Tools: former data can be extracted only with the `unyaffs` tools

EEP – EXPERIMENTS ANALYSIS

- Duration of the process and load which was used.

Load	Contacts & Calls	SMS & MMS	Bookmarks
Medium	120	250	45
High	222	591	93



CONCLUSION

- Classification and application of the Anti-Forensics techniques to Mobile Environment
- Proposed some possible instances have been fully automated by AFDroid
- Designed and performed experiments proving the AFDroid features

FUTURE WORK

- Improving AFDroid application that has been developed:
  - …to notice the capability to selectively choose the target evidence.
- Instantiating Anti-Forensics to other operating systems:
  - Windows Mobile, Symbian, etc…