Towards Secure Deletion on Smartphones

Michael Spreitzenbarth

GI-Sicherheit
6th October 2010
Introduction

Background: Wear Leveling

Towards Secure Deletion on Symbian OS

Conclusion & Future Work
Introduction I

- smartphones are increasingly used as “mobile office“
- they are more frequently equipped with GPS for navigation purposes
- => smartphones store a lot of sensitive data
the second market (e.g. eBay) for smartphones is increasing

many manufacturers simply replace the phone with a refurbished one if it malfunctions

=> the deletion of data becomes more and more important
Obstacles

- Smartphones offer only a limited user-interface.
- The user’s activities are often restricted to interaction with preinstalled tools.
- Mobile devices typically use flash chips to store data and these components use a technique called „wear leveling“.
Wear Leveling I

- the content of flash chips can only be changed a limited number of times

- afterwards so much voltage or time is required to write on the cell that it becomes impractical to use it any further

- => wear leveling provides a method to distribute the access at times when it is detected that they are receiving significantly uneven use
Wear Leveling II

Only 200 blocks are reused: \[ \frac{10,000 \text{ cycles} \times 200 \text{ blocks}}{50 \text{ blocks per file} \times 6 \text{ files per hour} \times 24 \text{ hours per day}} \approx 278 \text{ days or } \leq 1 \text{ year} \]

All 4,096 blocks are evenly used: \[ \frac{10,000 \text{ cycles} \times 4,096 \text{ blocks}}{50 \text{ blocks per file} \times 6 \text{ files per hour} \times 24 \text{ hours per day}} \approx 5,689 \text{ days or } > 15 \text{ years} \]
Wear Leveling III

- the intention of this process is even wear of the storage space

- according to Symbian[6] and Samsung[5], Nokia uses in its current device series Samsung OneNAND storage

- this storage has a special modification of the known wear leveling techniques [2]
Wear Leveling Process

1. System Start

2. Command from smartphone OS

3. Write on that block

4. Comparison of the deletion count of each block

5. Is the deletion marker of the actual block higher than the one of the first block in the garbage queue?

   - No: Erase the first block of the garbage queue
   - Yes: Update garbage queue

6. Write on the block from the garbage queue

7. Put actual block into the garbage queue
Towards Secure Deletion on Symbian Smartphones

- we developed a tool that helps to securely delete data on Symbian driven smartphones
- the tool is named „SecDel“ and is written in Python
Overview of the Tool

- SecDel currently possesses the ability to delete SMS messages, telephone directories, as well as calendar entries.
- It contains an update-function, which allows the user to load modules via the Internet.
- We have included a remote service function to run the deletion process remotely.
The Deletion Process

- a list of all entries in the telephone directory is created
- the user can choose the entry he wants to delete
- SecDel overwrites every part of this entry with a maximum amount of “X”
- SecDel deletes this entry with the help of a system API call
Some Impressions
all operations and testings have been realized on a Nokia E90 with Symbian 9.2 and a Nokia N70 with Symbian 8.1

flasher-tools like Twister Box will not work

for verification only a few technical possibilities exist
The Software Agent I

Local Execution = Collection

Target Smartphone

Storage

Forensic Workstation
The Software Agent II

- fast and easy to use
- is based on system API calls
- automated process
- offers analysis on a higher level
Desoldering the Chip

- high technical knowledge and equipment is needed
- time-consuming
- offers in depth analysis of the flash chip
Evaluation II

=> due to the fact that we did not have the technical possibilities to desolder the flash chip, we chose the software agent for verification
we used the software agents MIAT\textsuperscript{[1][4]} and Panoptes\textsuperscript{[7]}

\Rightarrow as a result these tools are not able to restore any deleted data

\Rightarrow the most perfect solution for the verification would be the desoldering of the flash storage chip and the subsequent direct analysis of this element
we presented a tool which tries to delete personal data in a more secure way
the tool first overwrites the data with garbage and deletes this garbage afterwards
unfortunately we were not able to thoroughly verify the deletion
we could prove with the help of tools like Panoptes[7] and MIAT[1][4]
Future Work

- optimization of the evaluation procedures (e.g. desoldering of the flash chip)
- transferring SecDel to Android OS
- forensic analysis of the Android OS and Android file system (YAFFS2)
Thank you very much for your Attention

- Michael Spreitzenbarth
- University of Mannheim
- Laboratory for Dependable Distributed Systems
- 68131 Mannheim - Germany
- spreitzenbarth@informatik.uni-mannheim.de
References I


References II


