Introduction to being a Privacy Detective
Investigating and Comparing Potential Privacy Violations in Mobile Apps using Forensic Methods

Stefan Kiltz, Robert Altschaffel, Thorsten Lucke, Jana Dittmann

Email: Robert.Altschaffel@iti.cs.uni-magdeburg.de

Otto-von-Guericke University
Magdeburg, Germany
• Research Assistant in Research Group Multimedia and Security, Otto-von-Guericke-University of Magdeburg
• Research interests:
  – Computer Forensics
  – Automotive IT
  – ICS (Industrial Control Systems)
  – Network Analysis
  – Data Protection/Privacy
• Broad range of publications on these research subjects
• Research group at the Otto-von-Guericke University Magdeburg, Germany
• Research fields and interests
  – Computer security, privacy, data sovereignty
  – Security in Automotive IT and Industrial Control systems (ICS)
  – Forensics (Desktop IT, crime scene, Automotive IT, Industrial Control Systems)
  – Watermarking and Steganography
  – Biometrics
• https://omen.cs.uni-magdeburg.de/itiamsl/deutsch/home/index.html
Outline

• Introduction
• Fundamentals
  – Methods to identify data flows and evaluate the privacy violations in apps
  – Computer forensics
• Structured approach to investigate and compare potential privacy violations in websites and apps
  – System landscape
  – Comparison of methods within the forensic framework
  – Visualization of examination results
• Case Study
  – Building a test environment
  – Test of different apps and results
• Conclusion
  – Future Work
Introduction

• Privacy and data protection are relevant topics (see Article 5 of the GDPR [1])
• Privacy is endangered by data flows, some of them undisclosed
  – used by third parties to identify customers, create profiles, send targeted advertisement
  – In addition, these data flows use unnecessary resources (e.g., CPU power, bandwidth, energy) without benefit to the user
• Discussion whether certain data flows violate the right of privacy of an user relies on legal background and a review of the relevant laws
• We provide a technical identification of said data flows
• Aim of this paper is to support privacy and data protection by providing means to identify data flows caused my mobile app(lications)
• This enables some degree of data sovereignty

Introduction – Data flows

- Four different, discernible types of data flows
- Differentiating a. between First Party (service provider) and Third party (other entities) and b. necessary and not necessary for the functionality of the app
  - Data flow to the service provider necessary for the functionality of the app (DF$_{fp.req}$)
  - Data flow to the service provider not necessary for the functionality of the app (DF$_{fp.nrq}$)
  - Data flow to a third party necessary for the functionality of the app (DF$_{tp.req}$)
  - Data flow to a third party not necessary for the functionality of the app (DF$_{tp.nrq}$)
- We refer to any data flow not necessary to provide the functionality intended by the user as a tracker
Fundamentals – Tracker identification

- Two principal approaches
  - **Static Analysis**
    - Investigating the binary representation of the app for known patterns (signatures)
    - Requires these signatures (including regular updates)
    - Examples: Exodus Privacy [1], Exodus Standalone [2], AppChecker [3]
  - **Dynamic Analysis**
    - App is executed and the communication behavior observed and analyzed
    - Requires knowledge in network analysis
    - Example: Wireshark [4]

[1] https://exodus-privacy.eu.org/2
Fundamentals – Tracker identification

- Some properties of the mobile domain impact tracker identification
  - Prop1: large amount of background processes
  - Prop2: very low control over operating system
  - Prop3: standardization of development tools
  - Prop4: reliance on system functions
  - Prop5: apps contain a manifest (containing information about requested system permissions)
  - Prop6: various variants
  - Prop7: App bundles

- Prop1 and Prop2 have a negative impact on the capabilities to perform dynamic analysis
- Prop5 eases the complexity of identifying permissions during static analysis
- Prop6 and Prop7 raise the difficulty of obtaining the correct binary for analysis in the first place.
Forensics describes a **scientific and systematic** approach for the reconstruction of events.

Forensic Process Models support the forensic process
- Structuring the process
- Making the process easier to describe and compare

In this paper we use the Forensic Process Model from [1] with additions from [2].

Of benefit for this paper is the structuring of the forensic process
- 6 Investigation Steps (phases of the process including a Strategic Preparation)
- 3 Data Streams (describing the origin of forensic data)
- 9 Data Types (describing how certain data is handled during the forensic process)

Aim: identify a **structured and comparable** approach to investigate trackers in mobile apps.


• **Data Streams** (based on [1] and [2])
  
  – Non-volatile Memory \((DS_T)\): Memory inside a computing unit which *maintains* its content after the unit is dis-connected from its respective power supply
  
  – Volatile Memory \((DS_M)\): Memory inside a computing unit which *loses* its content after the unit is disconnected from its respective power supply
  
  – Communication \((DS_N)\): All the data *transmitted* to other computing units via communication interfaces

---


Fundamentals - Computer Forensics

- **Data Types** (based on [1] and [2])
  - Hardware data (DT1): Data in a computing unit which is not, or only in a limited way, influenced by software
  - Raw data (DT2): A sequence of bits within the data streams of a computing systems not (yet) interpreted
  - Details about data (DT3): Data added to other data, stored within the annotated chunk of data or externally
  - Configuration data (DT4): Data which can be changed by software and which modifies the behavior of software and hardware, excluding the communication behavior
  - Network configuration data (DT5): Data that modifies system behavior with regards to communication
  - Process data (DT6): Data about a running software process within a computing unit
  - Session data (DT7): Data collected by a system during a session, which consist of a number of processes with the same scope and time frame
  - Application data (DT8): Data representing functions needed to create, edit, consume or process content relied to the key functionality of the system
  - Functional data (DT9): Data content created, edited, consumed or processed as the key functionality of the system

Structured approach to investigate and compare potential privacy violations in websites and apps

- Characteristics of investigation methods
  - Custody
    - Custody over the method
    - Custody over the examination item
    - either on-premises (the examiner has custody over this component) or off-premises
  - Examined Data stream
  - Type of examination
    - static examination
    - dynamic examination
Structured approach to investigate and compare potential privacy violations in websites and apps

- Comparison between methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Custody over Method</td>
<td>Off-premises</td>
<td>On-premises</td>
<td>On-premises</td>
</tr>
<tr>
<td>Custody over examination item</td>
<td>Off-premises</td>
<td>On-premises</td>
<td>On-premises</td>
</tr>
<tr>
<td>Data Stream</td>
<td>$DS_T$</td>
<td>$DS_T$</td>
<td>$DS_T$</td>
</tr>
<tr>
<td>Type of examination</td>
<td>Static</td>
<td>Static</td>
<td>Static</td>
</tr>
</tbody>
</table>

Structured approach to investigate and compare potential privacy violations in websites and apps

- All methods use a similar approach
  - Obtain the .APK (DT2)
  - Extract the .APK (DT2 -> DT2, DT3)
  - Extract list of hosts from binary (DT2 -> DT5)
  - Extract list of permissions from manifest (DT3 -> DT4)
  - Compare hosts to known signatures (DT5, external data -> DT5)
  - Compare permissions to known dangerous permissions (DT4, external data -> DT4)
  - Generate report (DT4, DT5 -> Report)

[1] https://exodus-privacy.eu.org/2
Structured approach to investigate and compare potential privacy violations in websites and apps

- Different ability to observe and document specific actions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Download .APK Extract .APK</td>
<td>DT2 DT2 -&gt; DT2, DT3</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Binary: Extract Hosts</td>
<td>DT2 -&gt; DT5 DT3 -&gt; DT4</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Manifest: Extract Permission</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Hosts: Compare</td>
<td>DT5, ext -&gt; DT5 DT4, ext -&gt; DT4</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Manifest: Compare</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Generate Report</td>
<td>DT4, DT5 -&gt; Report</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Visualization of examination results

• Large amount of results
• Goal is comparability
• Specific visualization required
  ➢ DNA-style graph including known trackers and permissions, also denoting the absence of these elements
Building a test environment

- .APKs were downloaded using emulators ([1] [2]) from the official store
- The SHA256sum was calculated to ensure integrity (and to compare if the correct version of the .APK is used)
- Exodus Standalone [3] and AppChecker [4] were installed locally
- Exodus Privacy [5] was used remotely
- The SHA256sum provided by Exodus Privacy allowed to confirm that all three methods examined the identical specimen

Tests of different apps and results

8 apps were tested using these three methods
Very few differences between Exodus Standalone [1] and Exodus Privacy [2] due to using the same engine while identifying trackers

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Corona-Warn 1.2.1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dropbox 194.2.6</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>GuitarTuna 6.4.0</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>Moodle</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Pixabay 1.1.3.1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>QR &amp; Barcode Scanner 2.1.32</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Shazam 10.38.0-200709</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Signal 4.69.4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Tests of different apps and results

- 8 apps were tested using these three methods
- No differences between all methods while identifying permissions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Corona-Warn 1.2.1</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Dropbox 194.2.6</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>GuitarTuna 6.4.0</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Moodle</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Pixabay 1.1.3.1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>QR &amp; Barcode Scanner 2.1.32</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Shazam 10.38.0-200709</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Signal 4.69.4</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
</tbody>
</table>

Tests of different apps and results – DNA graph
Summary

• Identification of data flows using a well-structured and comparable process in order to improve data sovereignty and data protection
• Three different methods of static analysis are employed in a test case containing 8 different applications identifying 42 trackers (20 unique) and 167 permissions (77 unique)
• Supporting users, developers, administrators, etc. in identifying unwanted data flows as a first step to prevent these flows
• Future work will include dynamic analysis

The research shown in this paper is partly funded by the European Union Project "CyberSec LSAOVGU-AMSL"
Summary

Links and References

GDPR
https://gdpr.eu/article-5-how-to-process-personal-data/ (30/10/2020)
Tools and Methods
https://exodus-privacy.eu.org/
https://github.com/Exodus-Privacy/exodus-standalone
https://github.com/Tienisto/AppChecker
https://www.wireshark.org
https://developer.android.com/studio/
https://www.genymotion.com

Thanks for your attention

The research shown in this paper is partly funded by the European Union Project ”CyberSec LS"