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

SECURWARE 2020
21.-25.11.2020
Valencia

Presenter: Robert Altschaffel

- 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

AMSL – Advanced Multimedia and Security Lab

- 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

[1] https://gdpr.eu/article-5-how-to-process-personal-data/ (30/10/2020)

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

AMSL Advanced Multimedia and Security Lab

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
[2] https://github.com/Exodus-Privacy/exodus-standalone
[3] https://github.com/Tienisto/AppChecker
[4] https://www.wireshark.org

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.

Fundamentals - Computer Forensics

- 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

[1] S. Kiltz, J. Dittmann, and C. Vielhauer, “Supporting Forensic Design – A Course Profile to Teach Forensics,” in Proc. 9th Int. Conf. on IT Security Incident Management & IT Forensics (IMF 2015). IEEE, 2015.
[2] R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann, “Digital Foren-sics in Industrial Control Systems,” in Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp 2019). Springer Nature Switzerland, 2019, pp. 128–136.

Fundamentals - Computer Forensics

- 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

[1] S. Kiltz, J. Dittmann, and C. Vielhauer, “Supporting Forensic Design – A Course Profile to Teach Forensics,” in Proc. 9th Int. Conf. on IT Security Incident Management & IT Forensics (IMF 2015). IEEE, 2015.
[2] R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann, “Digital Foren-sics in Industrial Control Systems,” in Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp 2019). Springer Nature Switzerland, 2019, pp. 128–136.

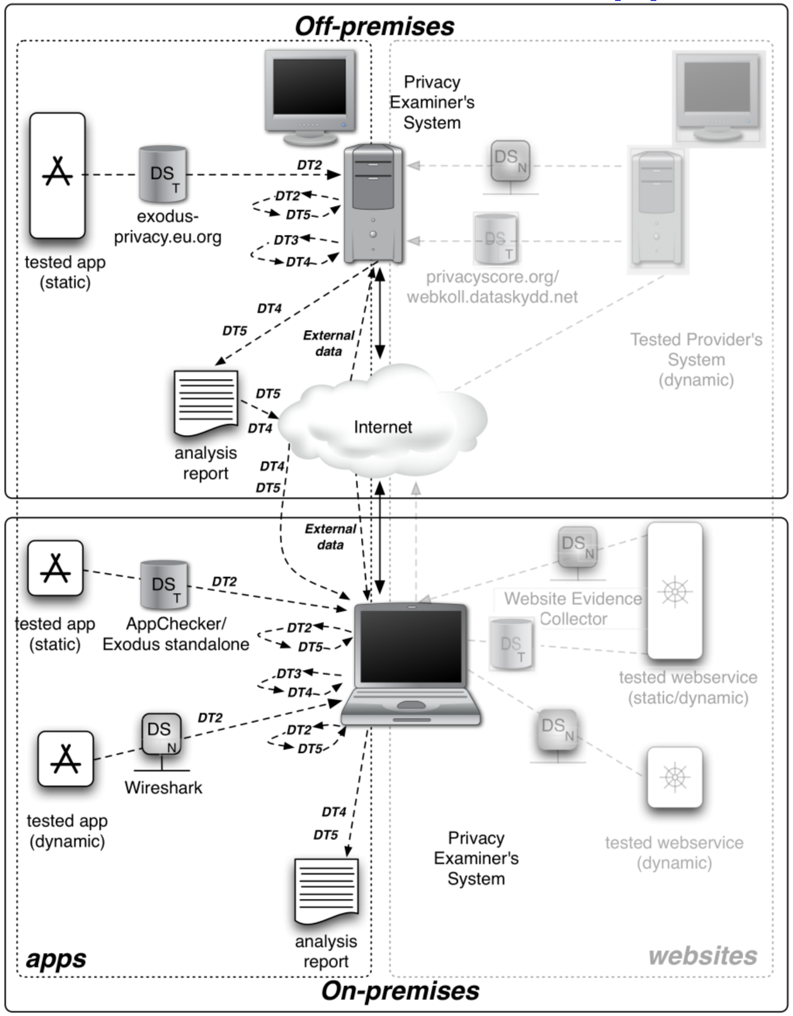
Fundamentals - Computer Forensics

- 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

[1] S. Kiltz, J. Dittmann, and C. Vielhauer, "Supporting Forensic Design – A Course Profile to Teach Forensics," in Proc. 9th Int. Conf. on IT Security Incident Management & IT Forensics (IMF 2015). IEEE, 2015.
[2] R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann, "Digital Foren-sics in Industrial Control Systems," in Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp2019). Springer Nature Switzerland, 2019, pp. 128–136.

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

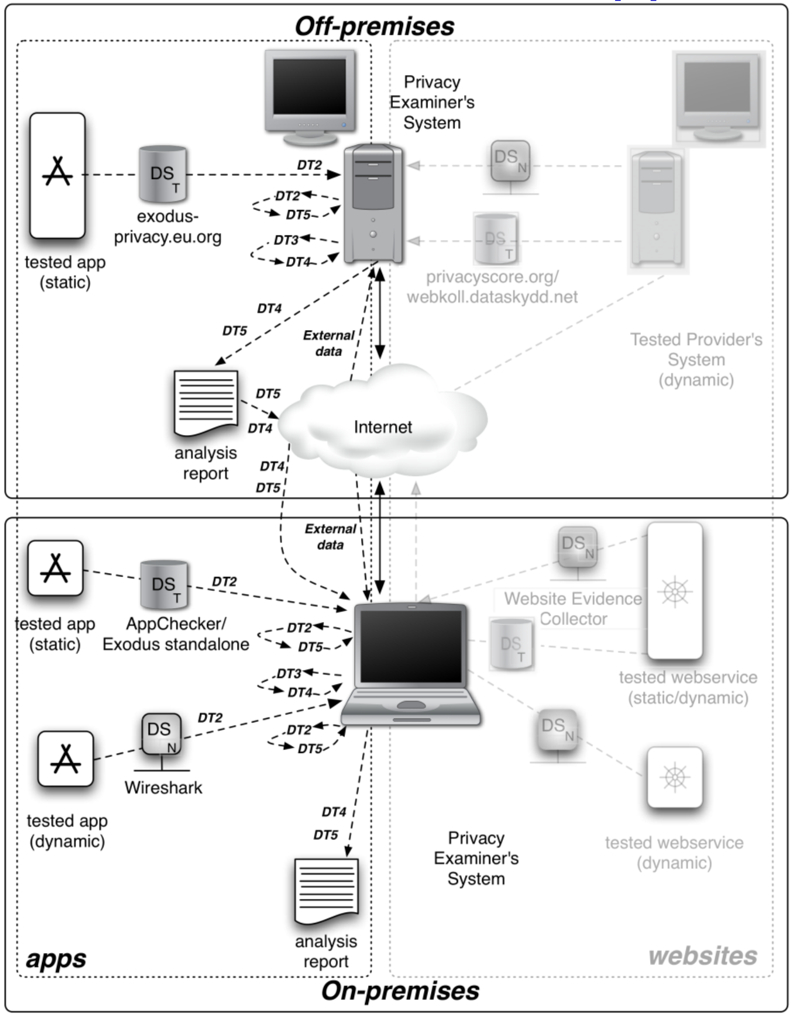
- Characteristics of investigation methods
  - Custody
    - Custody over the method
    - Custody over the examination item
    - either on-premises (the examiner has custody over this component) or off-premises
  - Examined Data stream
  - Type of examination
    - static examination
    - dynamic examination



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

- Comparison between methods

	Exodus Privacy [1]	Exodus Standalone [2]	AppChecker [3]
Custody over Method	Off-premises	On-premises	On-premises
Custody over examination item	Off-premises	On-premises	On-premises
Data Stream	DS_T	DS_T	DS_T
Type of examination	Static	Static	Static



[1] https://exodus-privacy.eu.org/2
[2] https://github.com/Exodus-Privacy/exodus-standalone
[3] https://github.com/Tienisto/AppChecker

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
[2] https://github.com/Exodus-Privacy/exodus-standalone
[3] https://github.com/Tienisto/AppChecker

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

- Different ability to observe and document specific actions

Internal Action	Data Types	Observable in Exodus Privacy [1]	Observable in Exodus Standalone [2]	Observable in AppChecker [3]
Download .APK Extract .APK	DT2 DT2 -> DT2, DT3	No No	Yes Yes	Yes Yes
Binary: Extract Hosts Manifest: Extract Permission	DT2 -> DT5 DT3 -> DT4	No No	Yes Yes	Yes Yes
Hosts: Compare Manifest: Compare	DT5, ext -> DT5 DT4, ext -> DT4	No No	Yes Yes	Yes Yes
Generate Report	DT4, DT5 -> Report	Yes	Yes	Yes

[1] https://exodus-privacy.eu.org/2
[2] https://github.com/Exodus-Privacy/exodus-standalone
[3] https://github.com/Tienisto/AppChecker

Stefan Kiltz, Robert Altschaffel, Thorsten Lucke, Jana Dittmann

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

[1] https://developer.android.com/studio13h
[2] https://www.genymotion.com
[3] https://github.com/Exodus-Privacy/exodus-standalone
[4] https://github.com/Tienisto/AppChecker
[3] https://exodus-privacy.eu.org/2

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

Application	Exodus Standalone [1]	Exodus Privacy [2]	AppChecker [3]	Common to all
Corona-Warn 1.2.1	0	0	1	0
Dropbox 194.2.6	5	5	4	2
GuitarTuna 6.4.0	10	10	11	9
Moodle	0	1	2	0
Pixabay 1.1.3.1	2	2	2	0
QR & Barcode Scanner 2.1.32	5	5	6	4
Shazam 10.38.0-200709	4	4	5	2
Signal 4.69.4	0	0	1	0

[1] https://github.com/Exodus-Privacy/exodus-standalone
[2] https://github.com/Tienisto/AppChecker
[3] https://exodus-privacy.eu.org/2

Tests of different apps and results

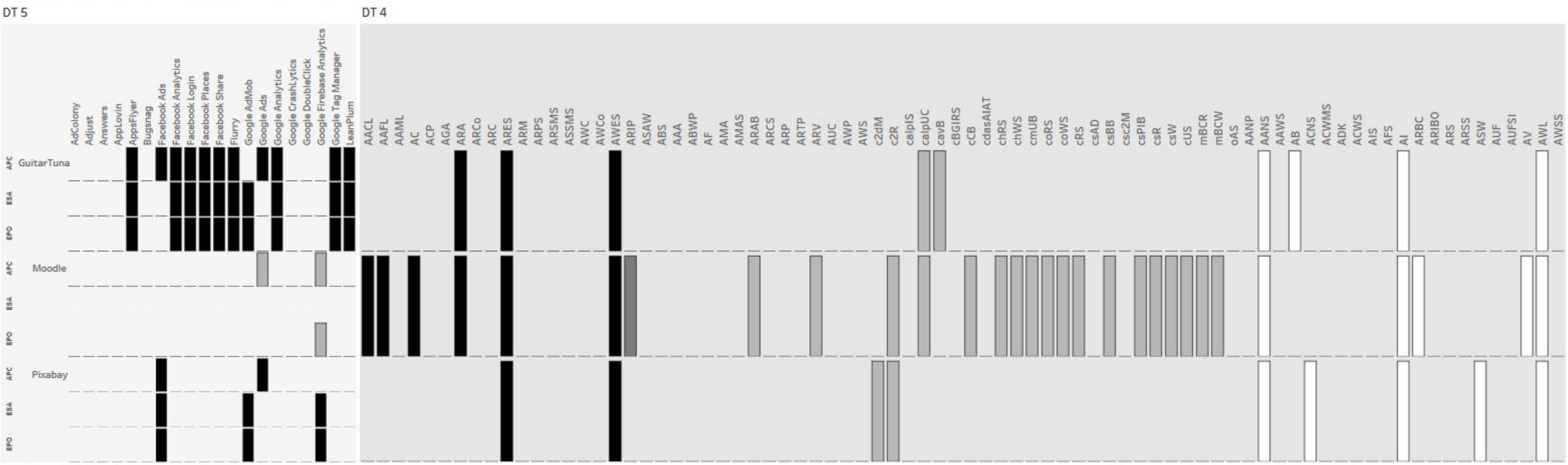
- 8 apps were tested using these three methods
- No differences between all methods while identifying permissions

Application	Exodus Standalone [1]	Exodus Privacy [2]	AppChecker [3]	Common to all
Corona-Warn 1.2.1	8	8	8	8
Dropbox 194.2.6	23	23	23	23
GuitarTuna 6.4.0	9	9	9	9
Moodle	30	30	30	30
Pixabay 1.1.3.1	9	9	9	9
QR & Barcode Scanner 2.1.32	13	13	13	13
Shazam 10.38.0-200709	14	14	14	14
Signal 4.69.4	65	65	65	65

[1] https://github.com/Exodus-Privacy/exodus-standalone
[2] https://github.com/Tienisto/AppChecker
[3] https://exodus-privacy.eu.org/2

Tests of different apps and results – DNA graph

Methods of EMID:
APC = AppChecker ESA = Exodus Privacy Standalone EPO = Exodus Provacy Online



AACL: android.permission.ACCESS_COARSE_LOCATION
AAFL: android.permission.ACCESS_FINE_LOCATION
AAML: android.permission.ACCESS_MEDIA_LOCATION
AANS: android.permission.ACCESS_NETWORK_STATE
AANP: android.permission.ACCESS_NOTIFICATION_POLICY
AAWS: android.permission.ACCESS_WIFI_STATE
AAA: android.permission.AUTHENTICATE_ACCOUNTS
AB: android.permission.BLUETOOTH
ABS: android.permission.BROADCAST_STICKY
ABWP: android.permission.BROADCAST_WAP_PUSH
ACP: android.permission.CALL_PHONE
AC: android.permission.CAMERA
ACNS: android.permission.CHANGE_NETWORK_STATE
ACWMS: android.permission.CHANGE_WIFI_MULTICAST_STATE
ACWS: android.permission.CHANGE_WIFI_STATE
ADK: android.permission.DISABLE_KEYGUARD
AF: android.permission.FLASHLIGHT
AFS: android.permission.FOREGROUND_SERVICE
AGA: android.permission.GET_ACCOUNTS
AIS: android.permission.INSTALL_SHORTCUT
AI: android.permission.INTERNET
AMA: android.permission.MANAGE_ACCOUNTS
AMAS: android.permission.MODIFY_AUDIO_SETTINGS
ARTP: android.permission.RAISED_THREAD_PRIORITY
ARAB: android.permission.READ_APP_BADGE
ARC: android.permission.READ_CALENDAR
ARCS: android.permission.READ_CALL_STATE
ARCo: android.permission.READ_CONTACTS
ARES: android.permission.READ_EXTERNAL_STORAGE
ARPS: android.permission.READ_PHONE_STATE
ARP: android.permission.READ_PROFILE
ARS: android.permission.READ_SMS
ARSS: android.permission.READ_SYNC_SETTINGS
ARBC: android.permission.RECEIVE_BOOT_COMPLETED
ARM: android.permission.RECEIVE_MMS
ARSMS: android.permission.RECEIVE_SMS
ARA: android.permission.RECORD_AUDIO
ARV: android.permission.RECORD_VIDEO
ARIBO: android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
ARIP: android.permission.REQUEST_INSTALL_PACKAGES
ASSMS: android.permission.SEND_SMS
ASW: android.permission.SET_WALLPAPER
ASAW: android.permission.SYSTEM_ALERT_WINDOW
AUC: android.permission.USE_CREDENTIALS
AUF: android.permission.USE_FINGERPRINT
AUFSI: android.permission.USE_FULL_SCREEN_INTENT
AV: android.permission.VIBRATE
AWL: android.permission.WAKE_LOCK
AWC: android.permission.WRITE_CALENDAR
AWCo: android.permission.WRITE_CONTACTS
AWES: android.permission.WRITE_EXTERNAL_STORAGE
AWP: android.permission.WRITE_PROFILE
AWS: android.permission.WRITE_SMS
AWSS: android.permission.WRITE_SYNC_SETTINGS
calpUC: com.anddoes.launcher.permission.UPDATE_COUNT
calpIS: com.android.launcher.permission.INSTALL_SHORTCUT
cavB: com.android.vending.BILLING
cdasAIAT: com.dropbox.android.service.ACCOUNT_INFO_ALARM_TRIGGER
c2R: com.google.android.c2dm.permission.RECEIVE
cBGIRS: com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE
cRS: com.htc.launcher.permission.READ_SETTINGS
cUS: com.htc.launcher.permission.UPDATE_SHORTCUT
cCB: com.huawei.android.launcher.permission.CHANGE_BADGE
chRS: com.huawei.android.launcher.permission.READ_SETTINGS
chWS: com.huawei.android.launcher.permission.WRITE_SETTINGS
cmUB: com.majeur.launcher.permission.UPDATE_BADGE
coRS: com.oppo.launcher.permission.READ_SETTINGS
c2dM: com.pixabay.pixabayapp.permission.C2D_MESSAGE
csR: com.sec.android.provider.badge.permission.READ
csW: com.sec.android.provider.badge.permission.WRITE
cs2M: com.shazam.android.permission.C2D_MESSAGE
csAD: com.shazam.android.preloadinfo.provider.ACCESS_DATA
csBB: com.sonyericsson.home.permission.BROADCAST_BADGE
csPIB: com.sonymobile.home.permission.PROVIDER_INSERT_BADGE
mBCR: me.everything.badger.permission.BADGE_COUNT_READ
mBCW: me.everything.badger.permission.BADGE_COUNT_WRITE
oAS: org.thoughtcrime.securesms.ACCESS_SECRETS

The permissions' protection levels:

Protection Level: Dangerous
Protection Level: Signature
No data/Not categorized
Protection Level: Normal

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

S. Kiltz, J. Dittmann, and C. Vielhauer, “Supporting Forensic Design – A Course Profile to Teach Forensics,” in Proc. 9th Int. Conf. on IT Security Incident Management & IT Forensics (IMF 2015). IEEE, 2015.

R. Altschaffel, M. Hildebrandt, S. Kiltz, and J. Dittmann, “Digital Foren-sics in Industrial Control Systems,” in Proceedings of 38th International Conference of Computer Safety, Reliability, and Security (Safecomp 2019). Springer Nature Switzerland, 2019, pp. 128–136.

GDPR

https://gdpr.eu/article-5-how-to-process-personal-data/ (30/10/2020)

Tools and Methods

https://exodus-privacy.eu.org/2

https://github.com/Exodus-Privacy/exodus-standalone

https://github.com/Tienisto/AppChecker

https://www.wireshark.org

https://developer.android.com/studio13h

https://www.genymotion.com

Thanks for your attention

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