ISEC TU Graz

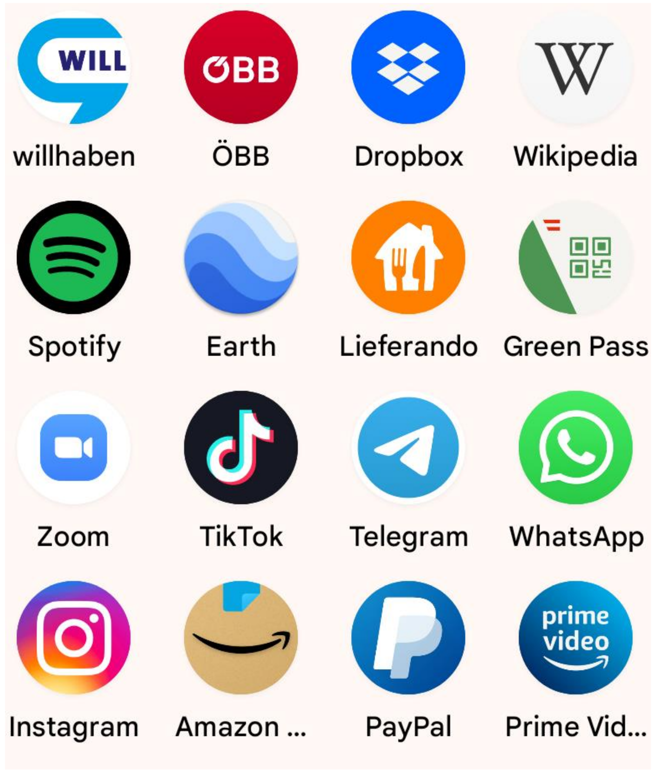
Practicals

Mobile Security 2026

Florian Draschbacher
florian.draschbacher@tugraz.at

Some slides based on material by **Johannes Feichtner**

…wants privacy



…

- *Am I talking to who I think I do?*
- *Does anyone tamper with my data?*
- *Who else gets access to my data?*
- *What information do they process, collect or share?*

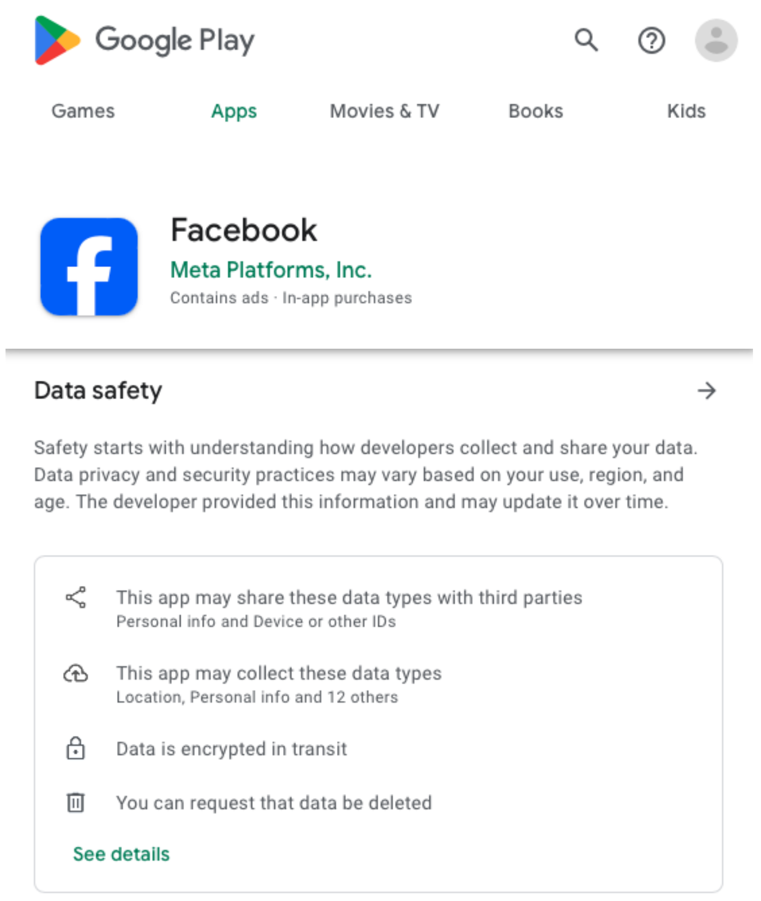
Data Safety Section on Google Play

- Permissions do not provide information on scope of data access
  - Is data processed, collected or shared?

- In 2022, Google introduced a **Data Safety Section** to Google Play

- Developer needs to disclose
  - What data does the app process, collect or share?
  - For what purpose is data shared?
  - Is disclosure optional?
  - Are security best practices followed?

Data Safety Section on Google Play


Google Play

Games Apps Movies & TV Books Kids

Facebook
Meta Platforms, Inc.
Contains ads · In-app purchases

Data safety

Safety starts with understanding how developers collect and share your data. Data privacy and security practices may vary based on your use, region, and age. The developer provided this information and may update it over time.

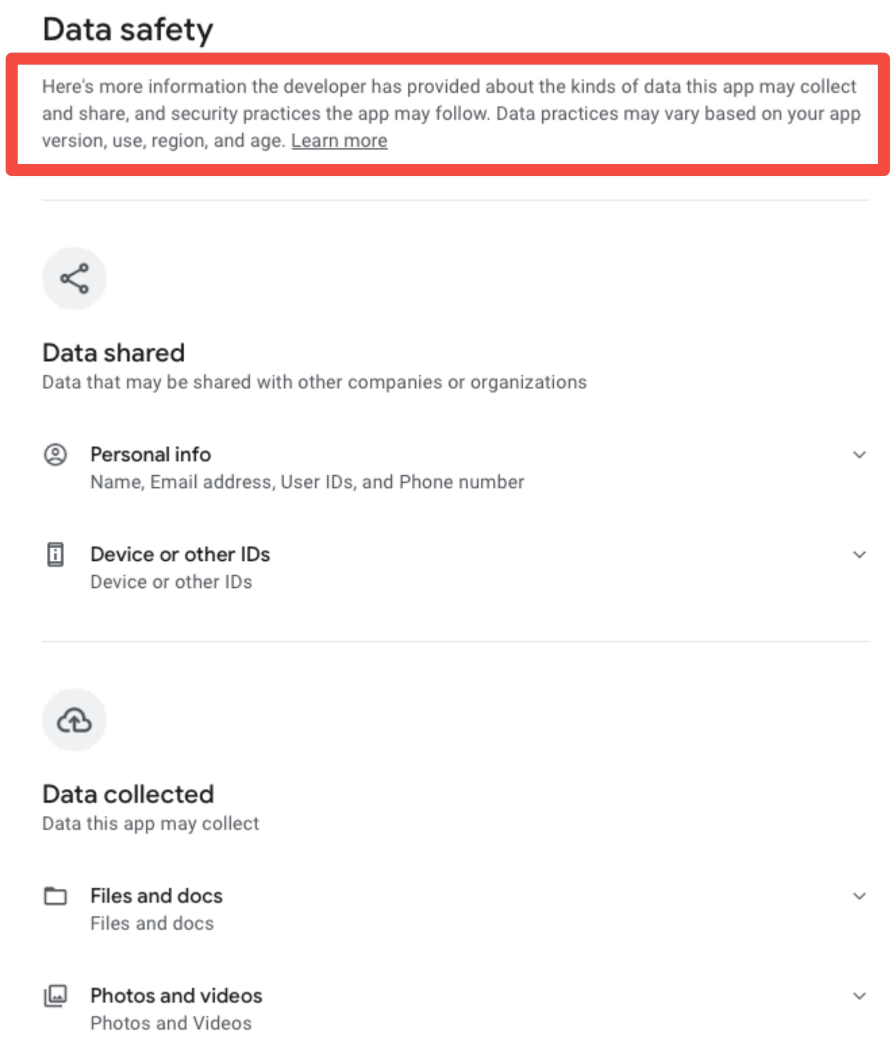
This app may share these data types with third parties
Personal info and Device or other IDs

This app may collect these data types
Location, Personal info and 12 others

Data is encrypted in transit

You can request that data be deleted

See details


Data safety

Here's more information the developer has provided about the kinds of data this app may collect and share, and security practices the app may follow. Data practices may vary based on your app version, use, region, and age. Learn more

Data shared
Data that may be shared with other companies or organizations

Personal info
Name, Email address, User IDs, and Phone number

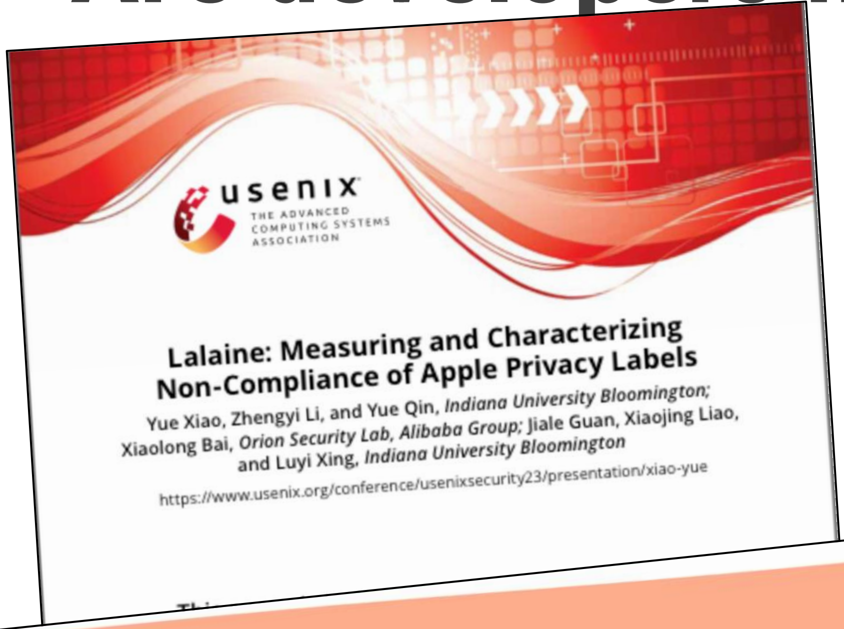
Device or other IDs
Device or other IDs

Data collected
Data this app may collect

Files and docs
Files and docs

Photos and videos
Photos and Videos

Are developers honest about their apps?

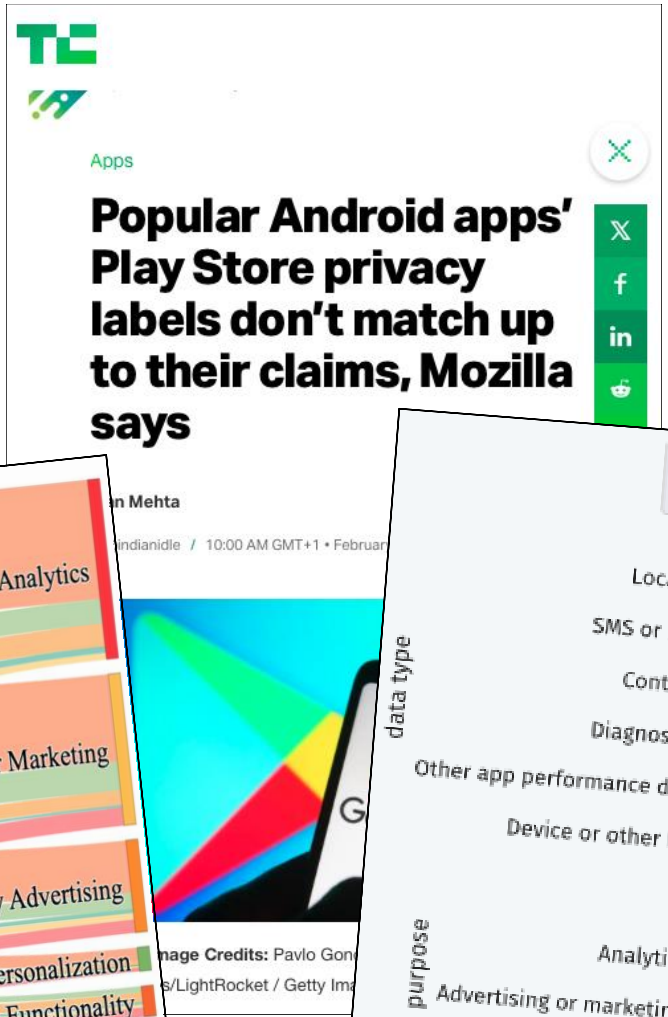


usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Lalaine: Measuring and Characterizing Non-Compliance of Apple Privacy Labels

Yue Xiao, Zhengyi Li, and Yue Qin, *Indiana University Bloomington;* Xiaolong Bai, *Orion Security Lab, Alibaba Group;* Jiale Guan, Xiaojing Liao, and Luyi Xing, *Indiana University Bloomington*

https://www.usenix.org/conference/usenixsecurity23/presentation/xiao-yue



TC

Apps

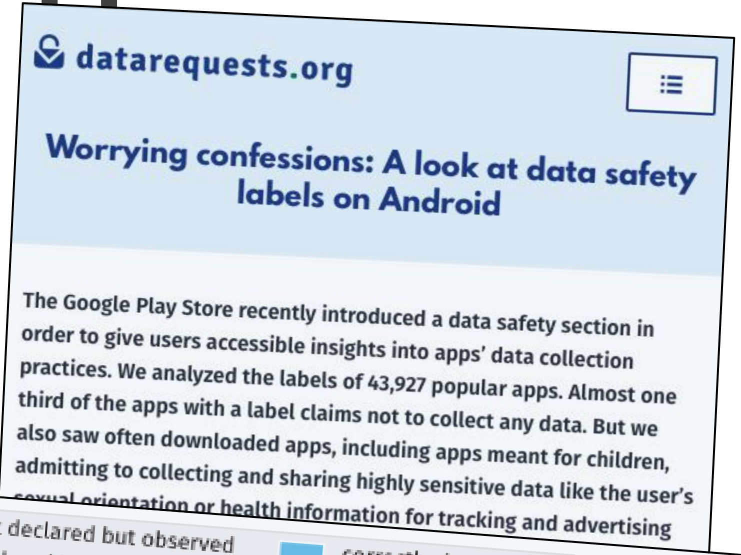
Popular Android apps' Play Store privacy labels don't match up to their claims, Mozilla says

n Mehta

indianidle / 10:00 AM GMT+1 • Februar

mage Credits: Pavlo Gon

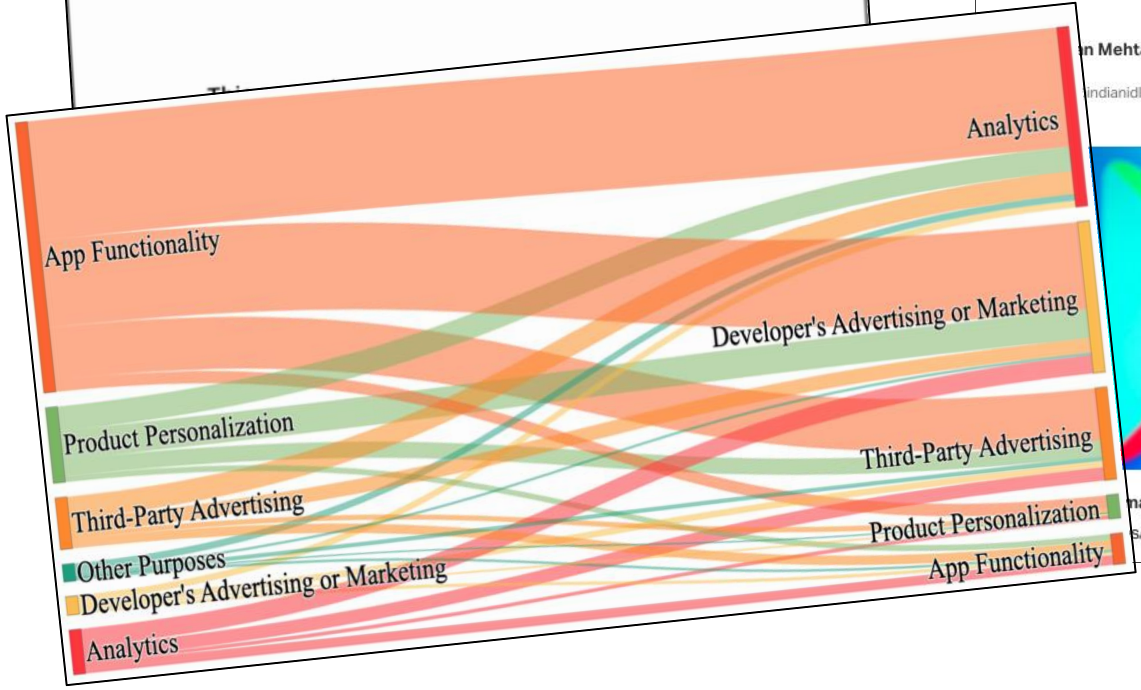
s/LightRocket / Getty Ima

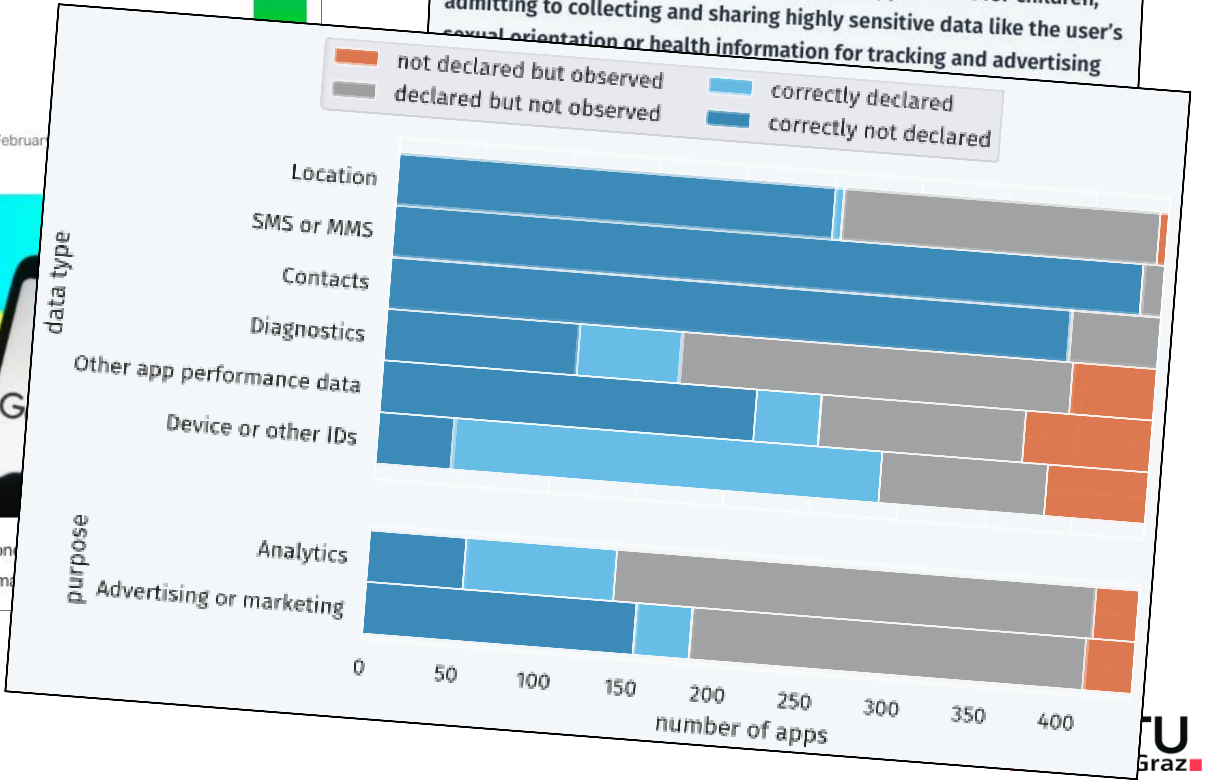


datarequests.org

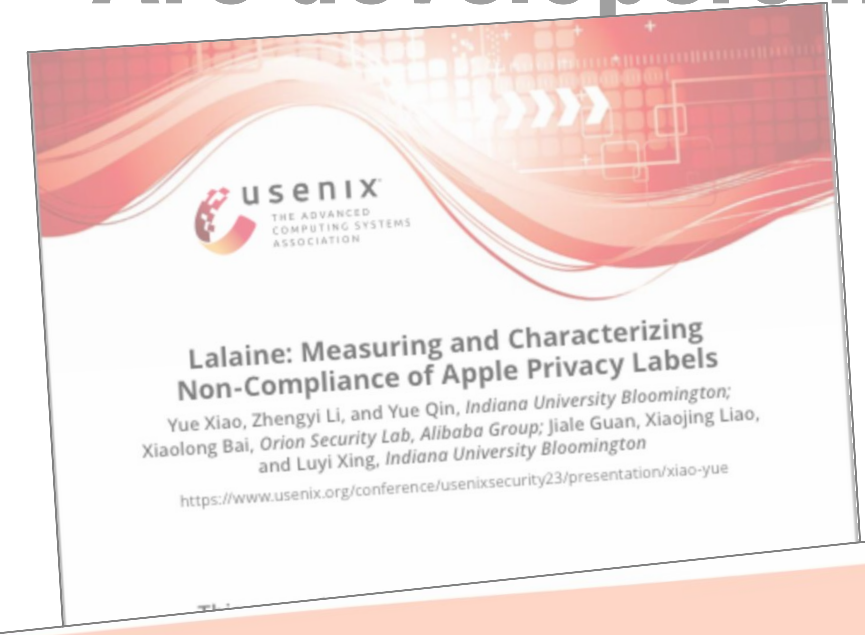
Worrying confessions: A look at data safety labels on Android

The Google Play Store recently introduced a data safety section in order to give users accessible insights into apps' data collection practices. We analyzed the labels of 43,927 popular apps. Almost one third of the apps with a label claims not to collect any data. But we also saw often downloaded apps, including apps meant for children, admitting to collecting and sharing highly sensitive data like the user's sexual orientation or health information for tracking and advertising





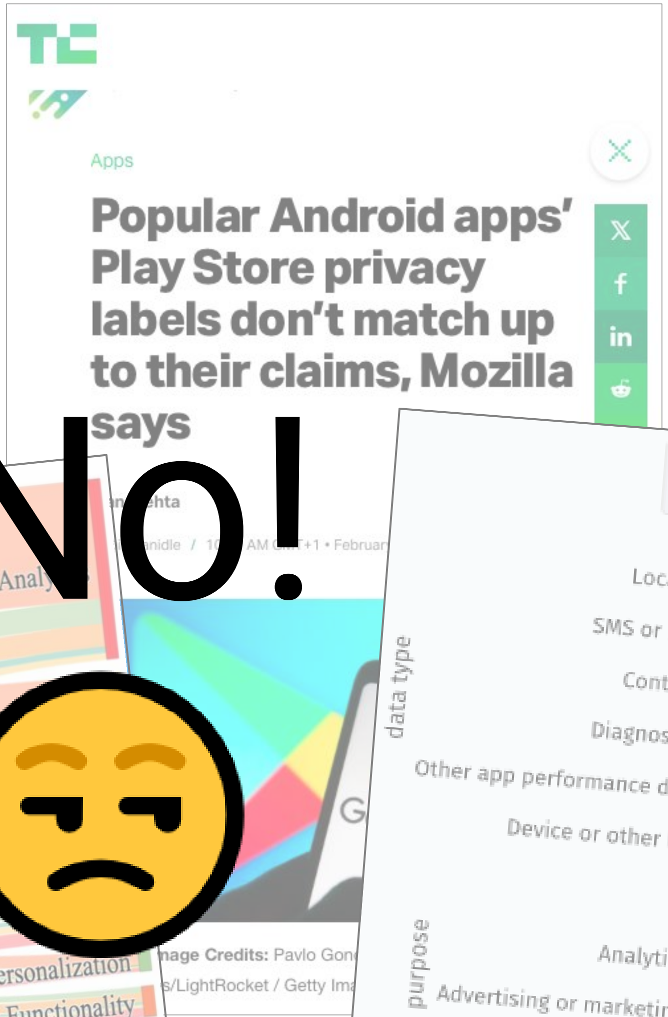
Are developers honest about their apps?


usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Lalaine: Measuring and Characterizing Non-Compliance of Apple Privacy Labels

Yue Xiao, Zhengyi Li, and Yue Qin, *Indiana University Bloomington;* Xiaolong Bai, *Orion Security Lab, Alibaba Group;* Jiale Guan, Xiaojing Liao, and Luyi Xing, *Indiana University Bloomington*

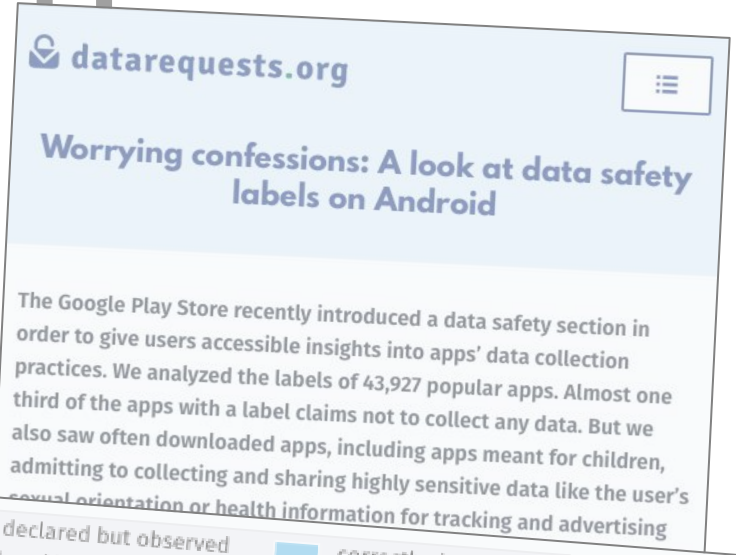
https://www.usenix.org/conference/usenixsecurity23/presentation/xiao-yue


TC

Apps

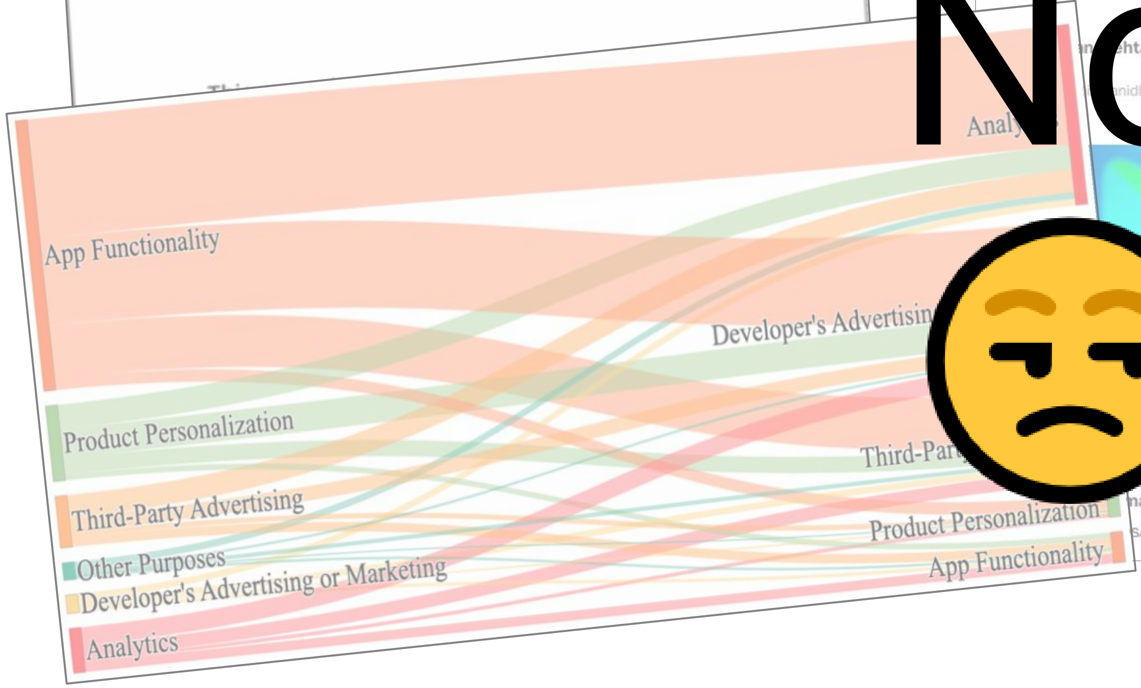
Popular Android apps' Play Store privacy labels don't match up to their claims, Mozilla says

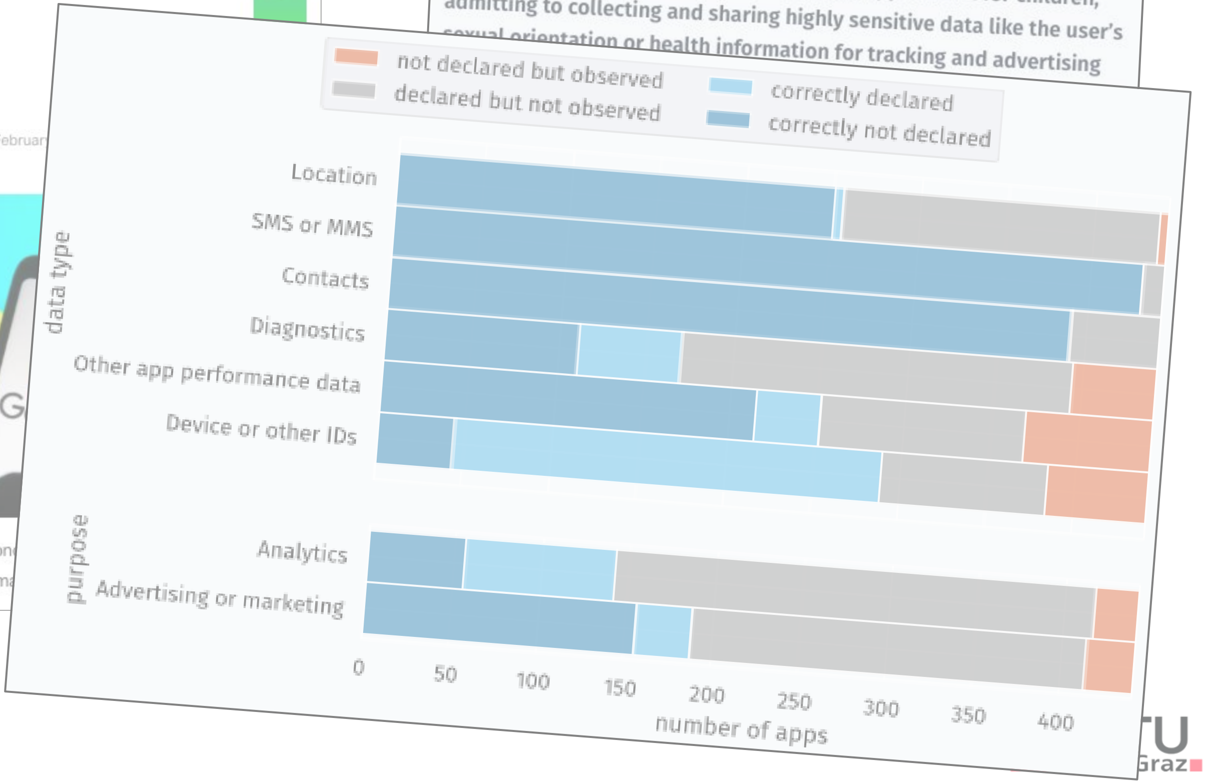
Image Credits: Pavlo Gon
s/LightRocket / Getty Ima


datarequests.org

Worrying confessions: A look at data safety labels on Android

The Google Play Store recently introduced a data safety section in order to give users accessible insights into apps' data collection practices. We analyzed the labels of 43,927 popular apps. Almost one third of the apps with a label claims not to collect any data. But we also saw often downloaded apps, including apps meant for children, admitting to collecting and sharing highly sensitive data like the user's sexual orientation or health information for tracking and advertising





No!

Your Task

Task 1

Analyse a set of **3 applications**

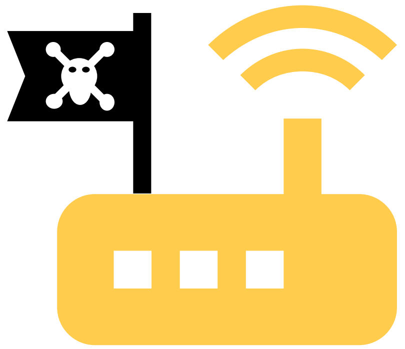
- Find out what data they transmit to their backend server(s)
- Check if their Data Safety Section is accurate

Roadmap for each app:

1. Carry out MITM attack to intercept backend communication
2. Analyze transmitted data
3. Compare with Data Safety Section
4. Write report of your findings

Grading of Task 1: Your result report

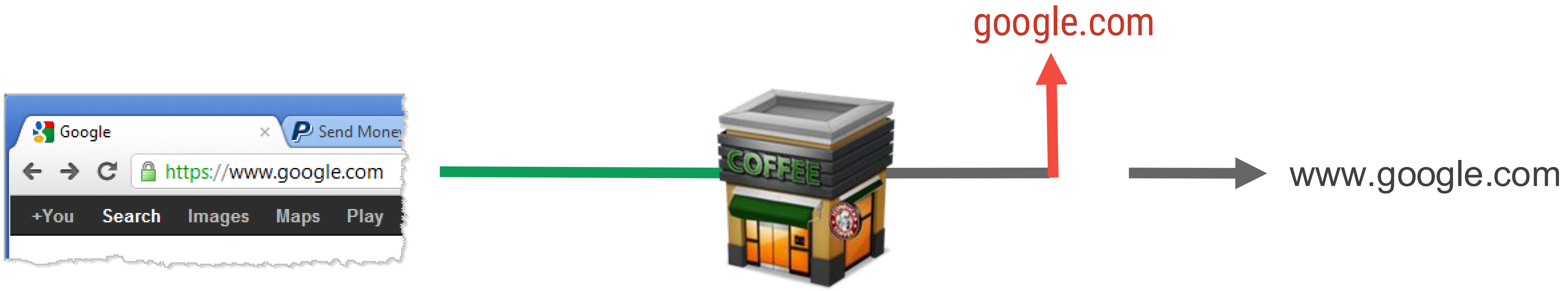
Recap: Man-in-the-middle



Picture: Google / Apache 2.0

Active attacker

Secretly relay (and possibly modify) traffic between client and server

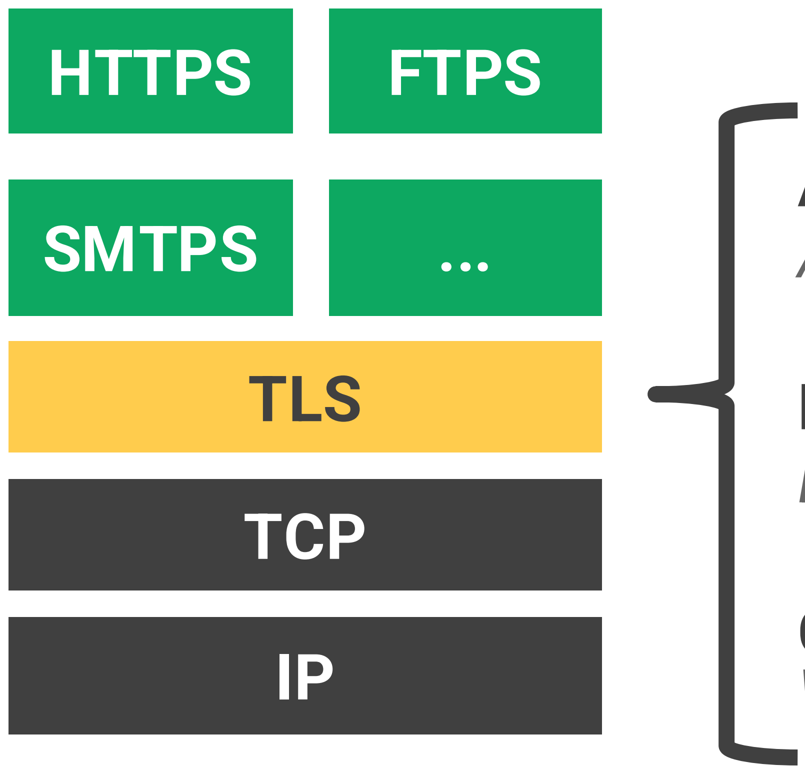


Picture: blaugrana-tez / CC BY-NC-ND

Client

Ideally does not notice anything (from an attacker's perspective)

Recap: Transport Layer Security



Problem: „Secure Identity“

Authentication
Am I talking to who I think I do?

Data integrity
Does anyone tamper with my data?

Confidentiality
Who else can see my conversation?

Problem: Key Exchange

Recap: Practical Defenses against MITM

- Use Transport Layer Security

- Validate server certificate chain
  - From server certificate to device-installed CA
  - Baseline of TLS security
  - Some developers disable validation for supporting self-signed certificates
    - Very bad idea!

- Implement certificate pinning
  - Hard-code the expected hash of the server certificate
  - Prevents attacks that involve state actors, malicious or compromised CAs

TLS on Android

- `SSLSocket` class for establishing secure TLS or SSL connection

- Validating certificate chain: `TrustManager`
  - Default: Trust CA in device trust store
  - Custom implementations may perform any validation logic (or none at all)

- Ensuring certificate hostname matches server hostname: `HostnameVerifier`
  - **Has to be invoked by code above SSLSocket**
  - **Developer's responsibility!**

HTTPS on Android

- Use Android's `HttpsURLConnection` class
  - By default: Secure `TrustManager` and `HostnameVerifier` (Details depend on Android version)
  - Possibility to use custom `TrustManager` and `HostnameVerifier`

- Use a third-party library such as OkHttp (built on top of `SSLSocket`)
  - Usually secure custom `TrustManager` and `HostnameVerifier`
  - Support self-signed certificates, certificate pinning, …

- Implement a custom HTTP stack on top of `SSLSocket`
  - Secure system-default `TrustManager`
  - **HostnameVerifier up to developer!**

Situation Pre-Android 7

```
Q: “Does someone know how to accept a self-signed certificate on Android?
    A code sample would be perfect.”
A: “Use the AcceptAllTrustManager”.

Q: “All I need to do is download some basic text-based and image files from
   a web server that has a self-signed SSL certificate...getting the SSL to
   work is a nightmare...”
A: “I found two great examples of how to accept self-signed SSL
   certificates, one each for HttpsURLConnection and HttpClient.”
```

[Source: Stackoverflow]

Applications

- Can overwrite certificate validation routines (system default: correct check)
- Self-signed certificates → used to require custom TrustManager
- Used to have to implement pinning on their own if wanted

Network Security Configuration (Android 7)

- XML-based system for configuring self-signed certificates and pinning
- These use cases no longer require custom validation code
- Default NSC: Don't trust user-installed CA certificates

However

- Even the NSC can be misconfigured
  - Trust user-installed CAs
- Some applications still use custom `TrustManagers` or `HostnameVerifiers`
  - **Overrides the NSC system altogether**

Source: developer.android.com

Task 1 – Detailed Steps (for each of the 3 apps)

1. Try to intecept app's traffic using proxy server
2. If any HTTP connections or insecure HTTPS

   ➔ Document this fact, go to step 5
3. Decompile app to find out why it didn't trust your CA
   – HTTP library, NSC, custom TrustManager?
4. Modify app to trust user-installed CAs
   – E.g. Recompile, resign, reinstall the app
5. Analyse the intercepted server communication
   – Is the Google Play Data Safety section accurate?
6. Document all findings in a scientific report

More details on assignment website

On the dark side…

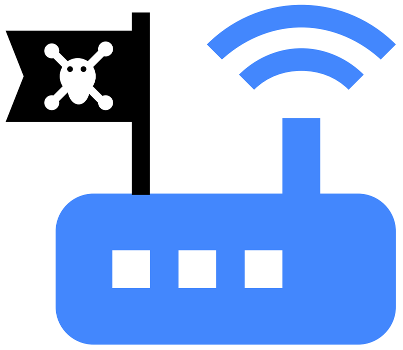
MITM attack tools

- mitmproxy.org, Fiddler, Proxyman, …

Decompiling and modifying Android apps

- JADX, Apktool
- Uber-APK-Signer
- A2P2 – Android Application Patching Pipeline
  https://extgit.isec.tugraz.at/fdraschbacher/a2p2
- Frida, Android-Unpinner

- Whatever way you choose for bypassing TLS pinning, document how it works!
  - Your report must prove you understand what is happening behind the scenes!


Picture: Google / Apache 2.0

ISEC TU Graz

Submission

- Submit **until 27.03.2026**:
  - Scientific report in PDF format
  - **Email to mobilesec.isec@tugraz.at**
- **Describe how** you analysed each of the applications
  - Text, app screenshots, excerpts from dumps, etc…
  - Provide reasoning for your approach
- **Describe** your findings
  - Is the communication protected as declared in the Data Safety section?
  - Is any data transmitted in conflict with the Data Safety section?
  - Any other interesting findings?

Reminder: Task 2

- Select a topic for assignment 2 until **20.03.2026**
- Plenty of topics to chose from on website
  - Or suggest your own!
- Groups of up to 3 people
  - But also possible to work on your own
- Send an email to mobilesec.isec@tugraz.at about group members and topic

Questions?