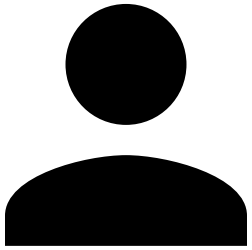
ISEC TU Graz

Assignment 1

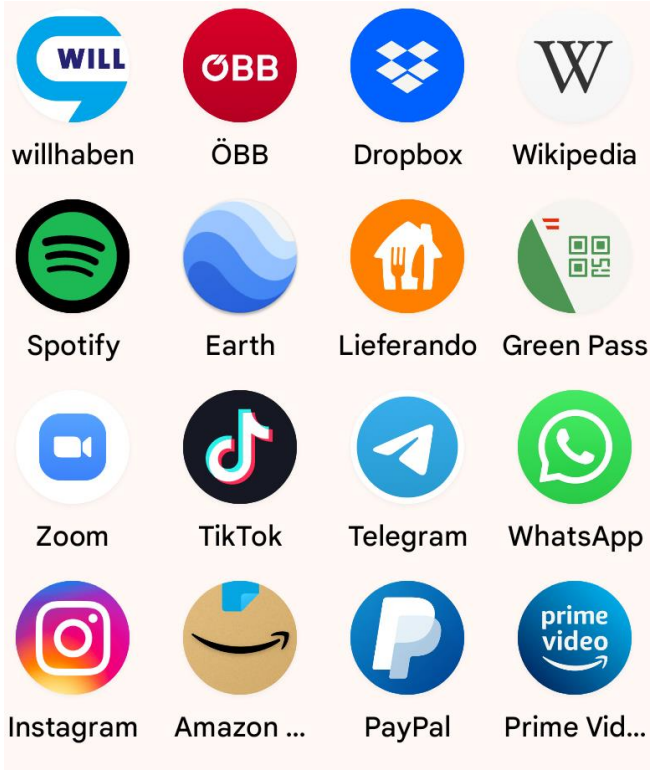
Mobile Security 2025

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?*

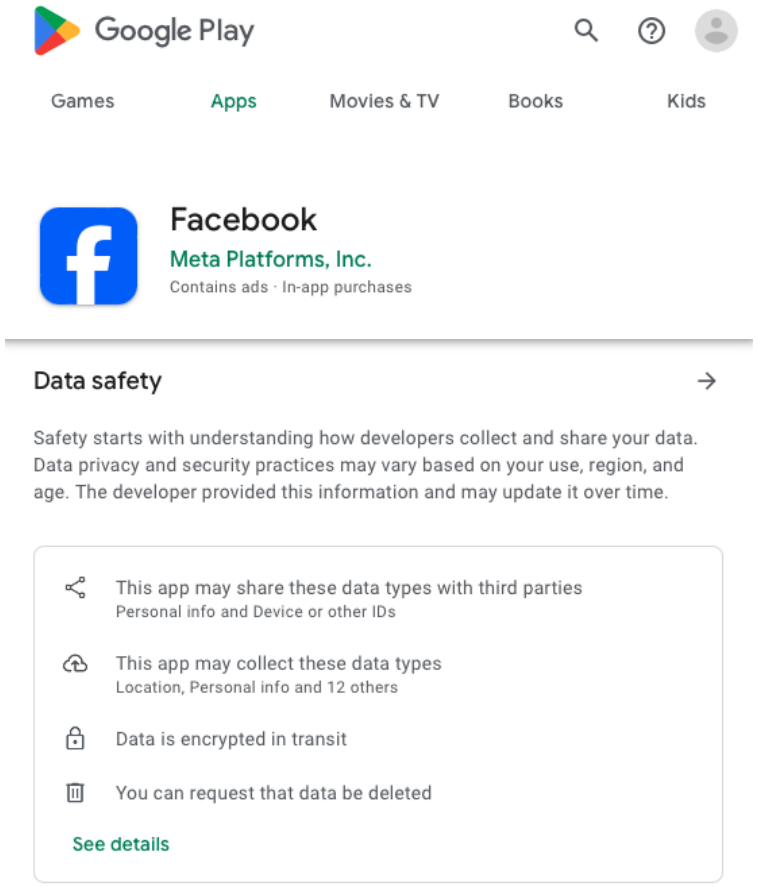
Data Safety Section on Google Play

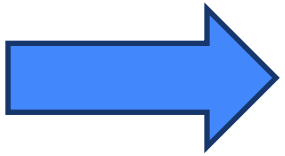
- Permissions do not provide information on scope of data access
  - Is data processed, collected or shared?

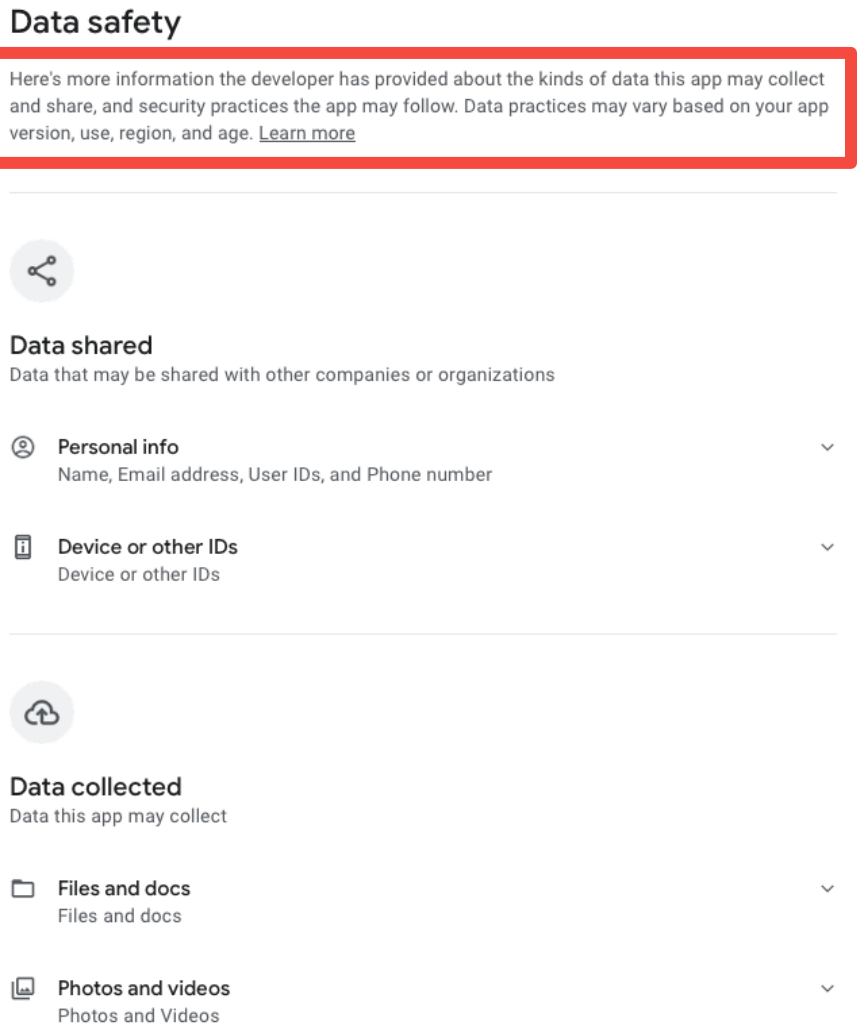
- In 2022, Google introduced a **Data Safety Section** to Google Play

- 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?

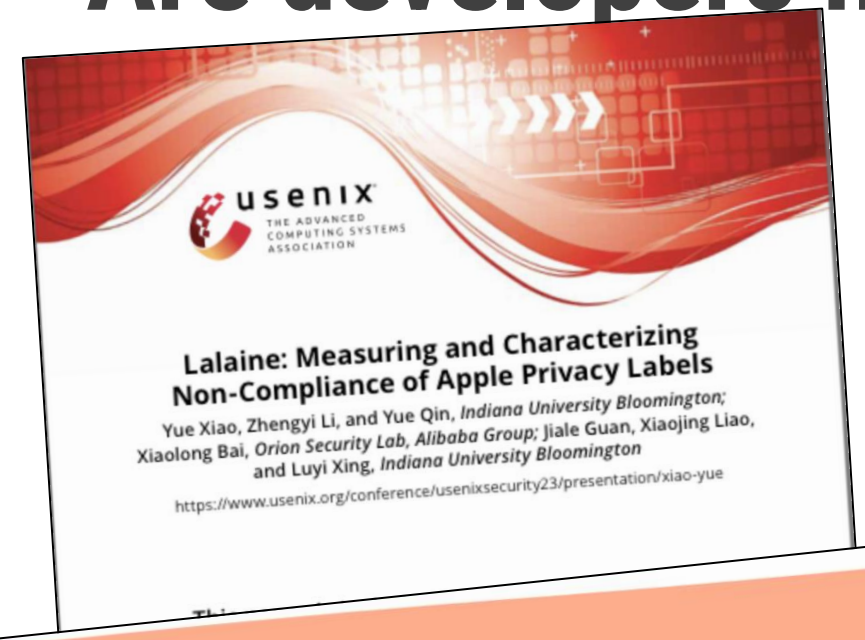
Data Safety Section on Google Play

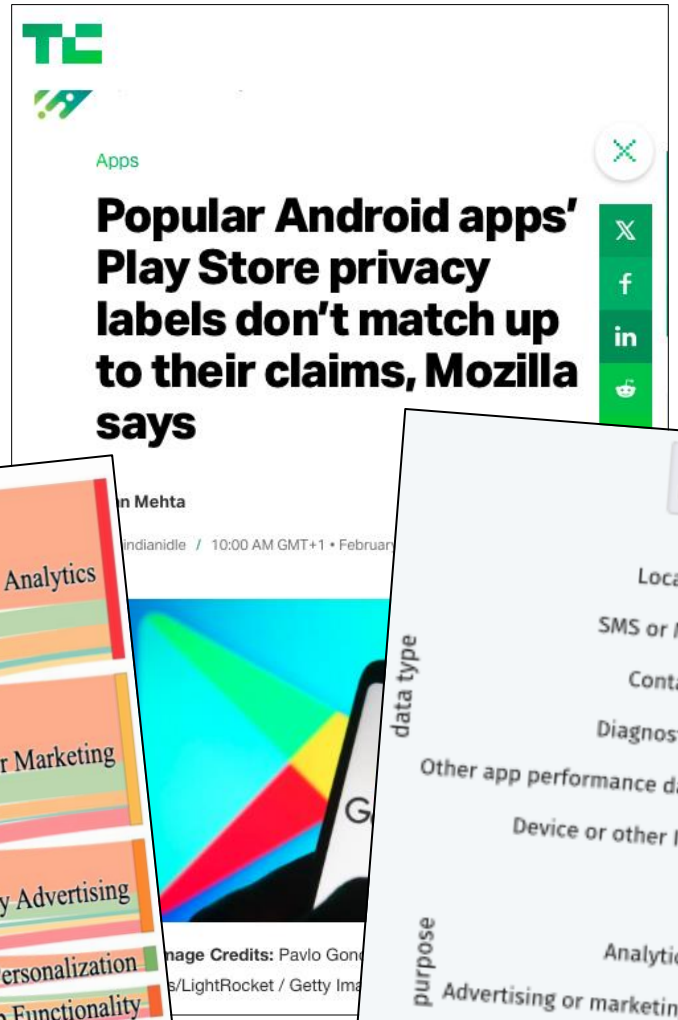


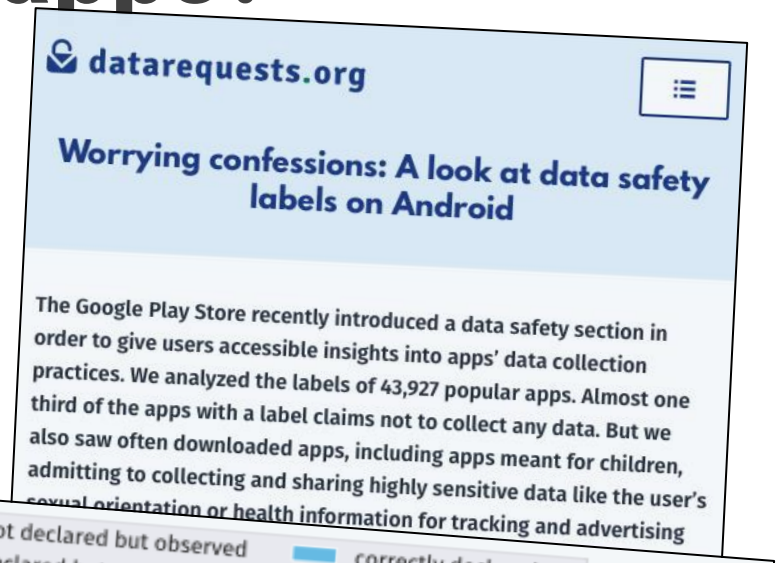


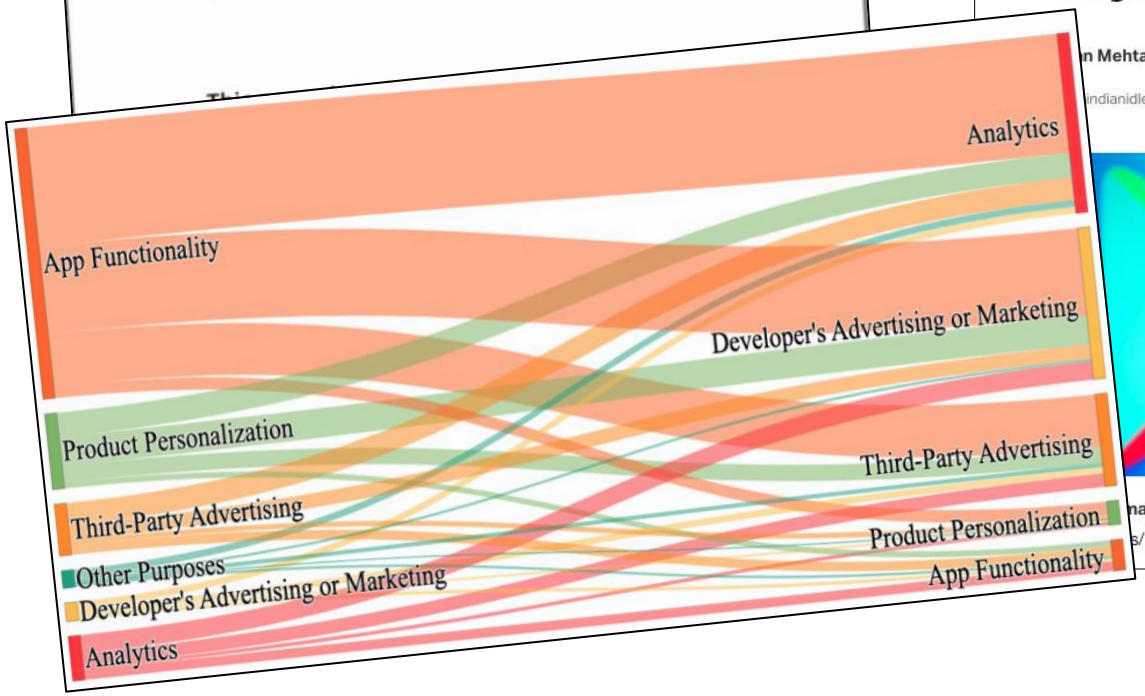


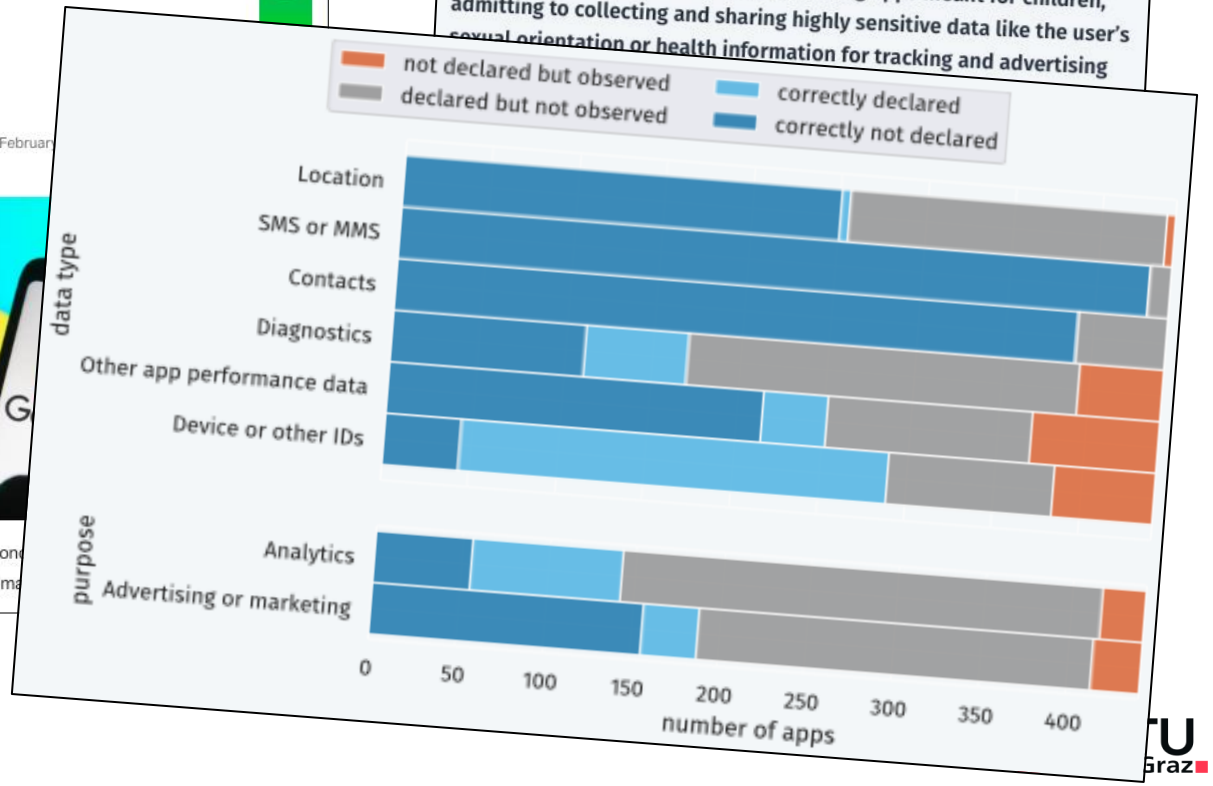
Are developers honest about their apps?



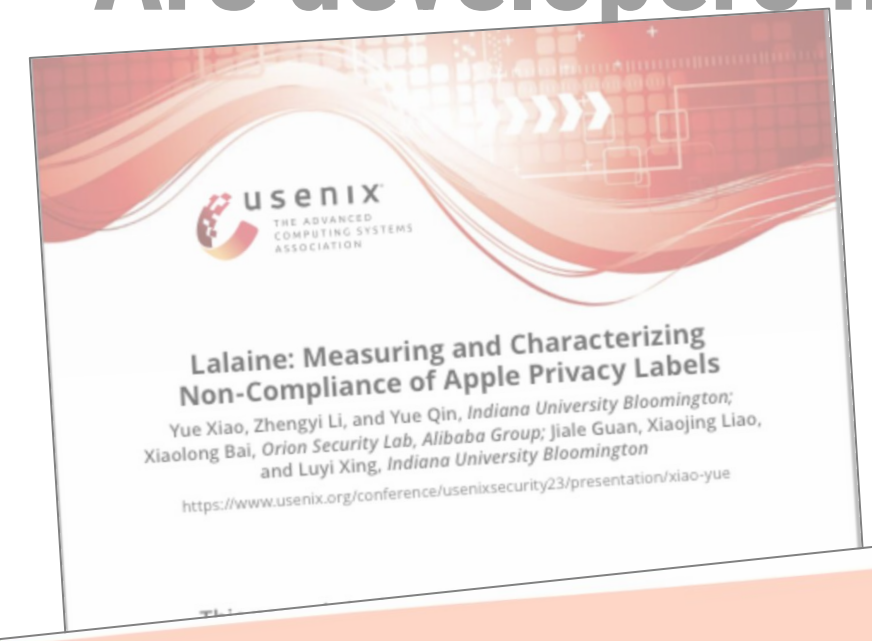


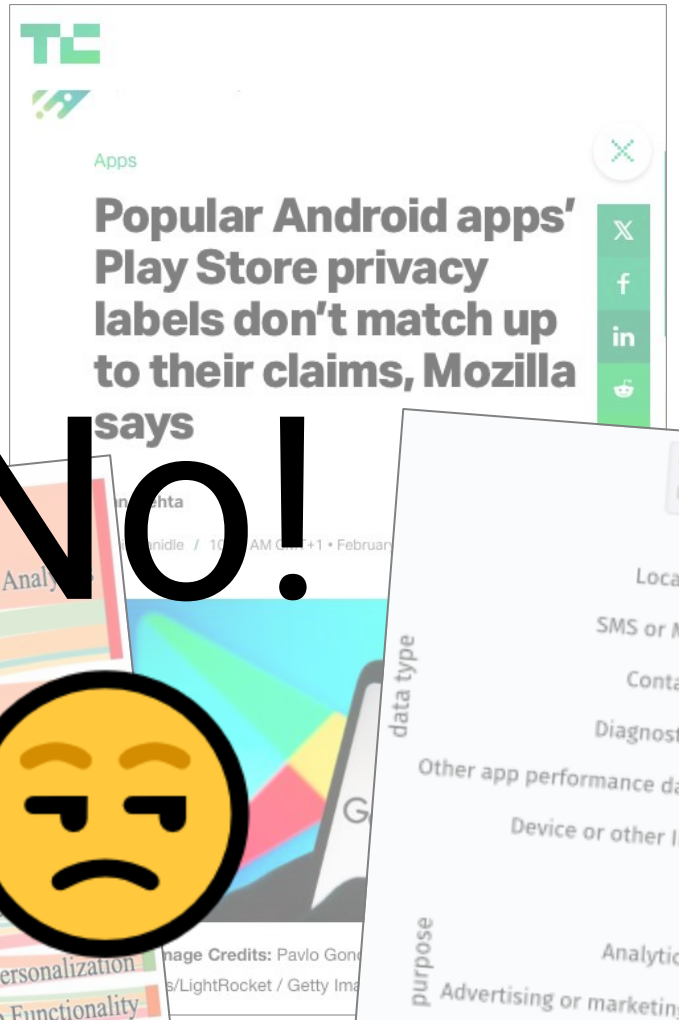


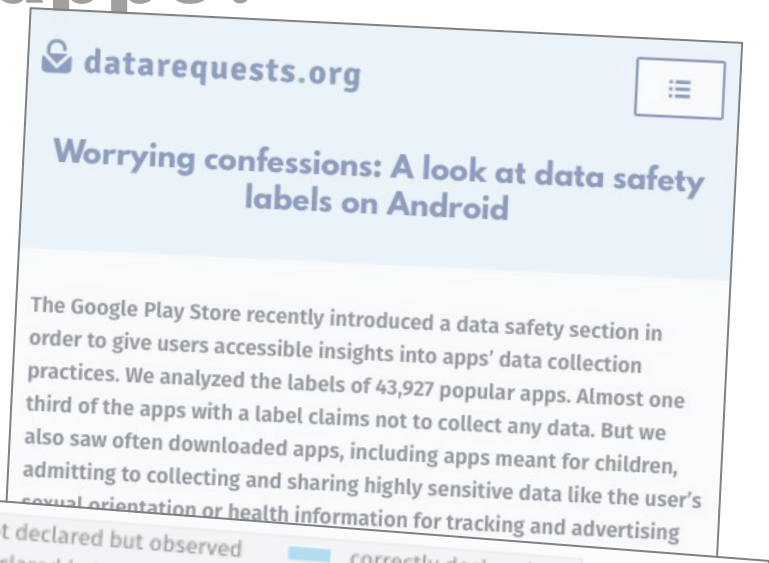




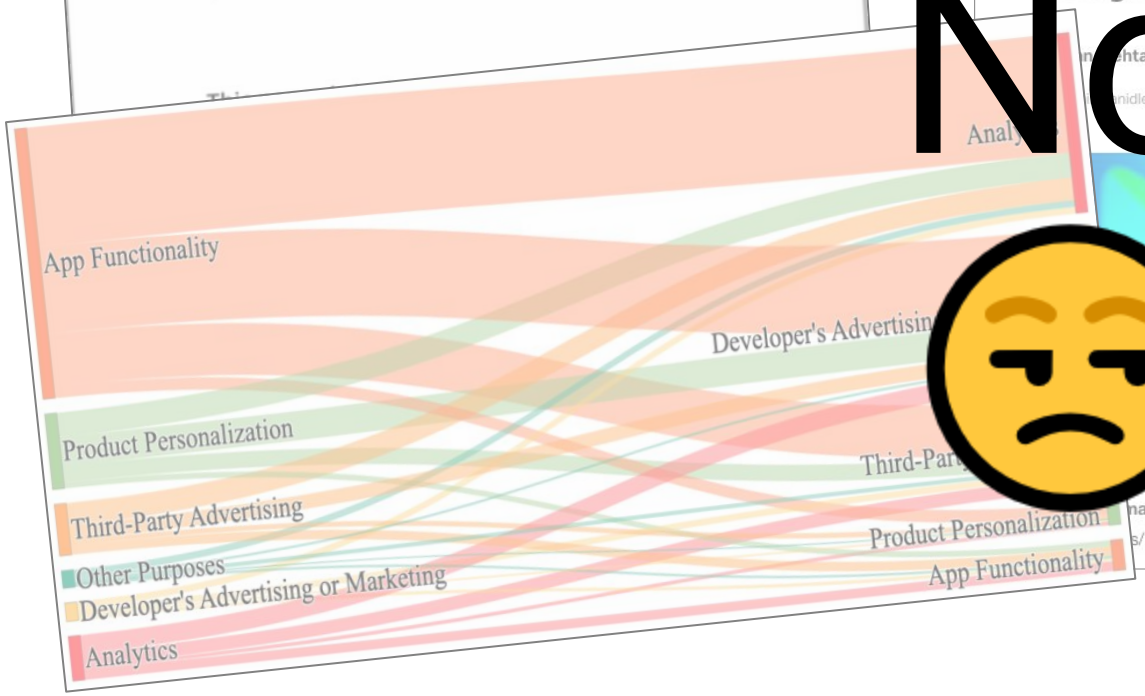
Are developers honest about their apps?

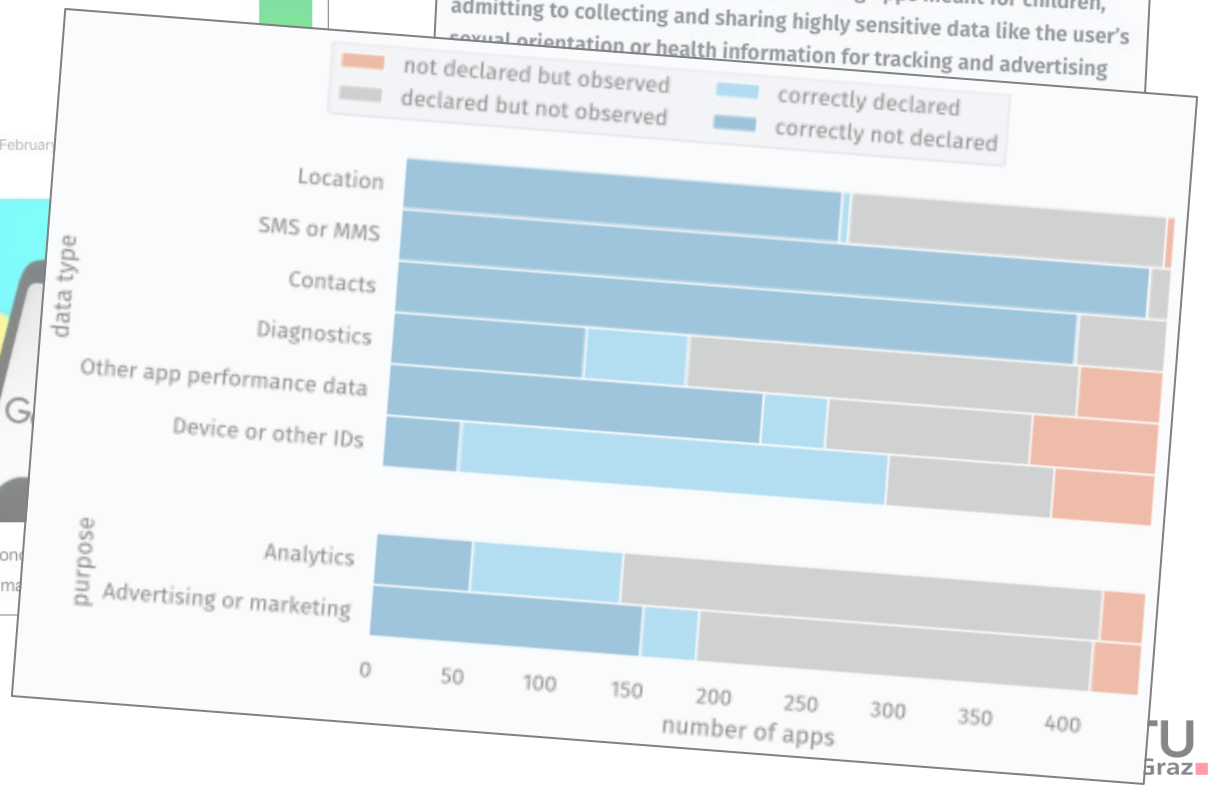






No!





Your Task

Task 1

Analyse a set of **3 applications**

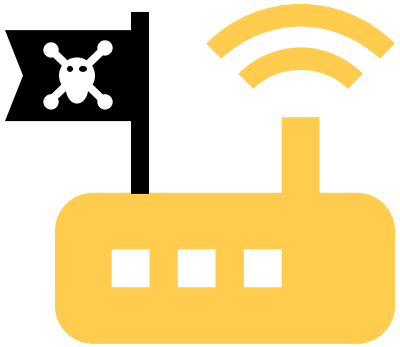
- Find out what data they transmit to their backend server
- Check if their Data Safety Section is accurate

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

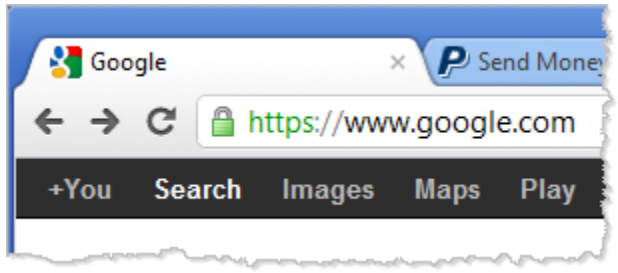
Recap: Man-in-the-middle



Active attacker

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

Picture: Google / Apache 2.0



google.com

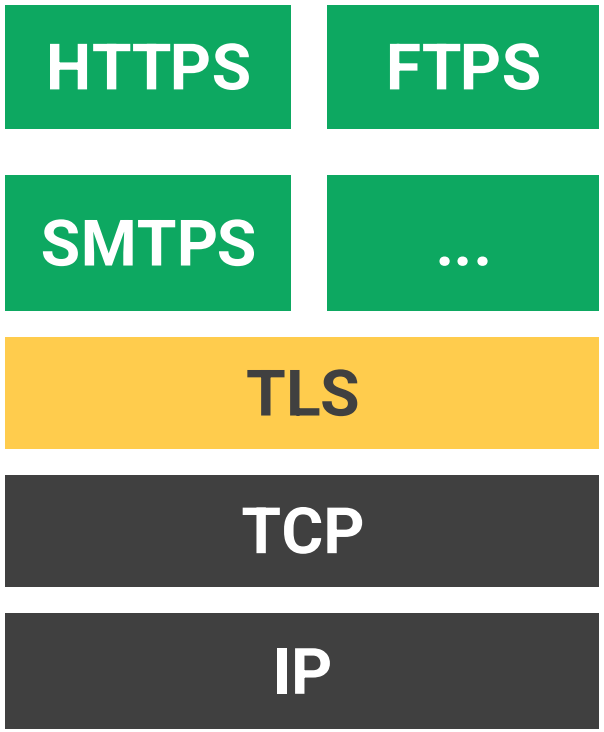
www.google.com

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
    - Involve users who installed additional CA certs to their device

TLS on Android

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

- Validating certificate chain: `TrustManager`
  - Default: Trust any CA installed on device
  - 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 how pinning is implemented
   - 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

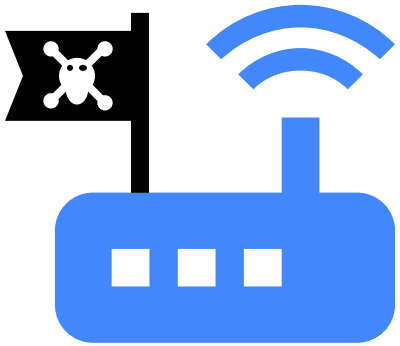
On the dark side…

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



Picture: Google / Apache 2.0

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

Submission

- Submit **until 28.03.2025**:
  - Scientific report in PDF format
  - **Email to mobilesec@iaik.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 **21.03.2025**
- **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@iaik.tugraz.at about group members and topic**

Questions?