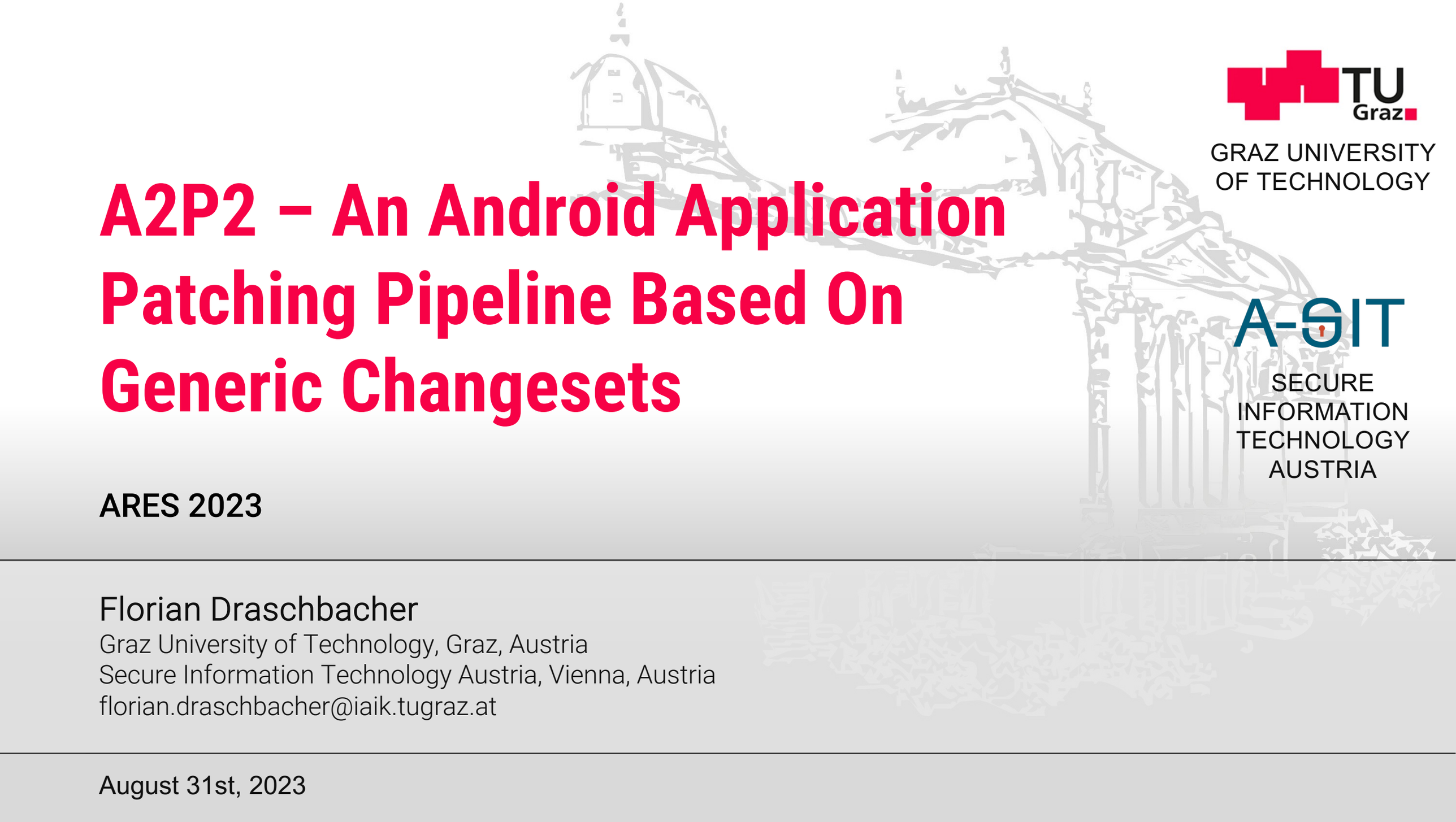


A2P2 – An Android Application Patching Pipeline Based On Generic Changesets

GRAZ UNIVERSITY OF TECHNOLOGY

A-SIT SECURE INFORMATION TECHNOLOGY AUSTRIA

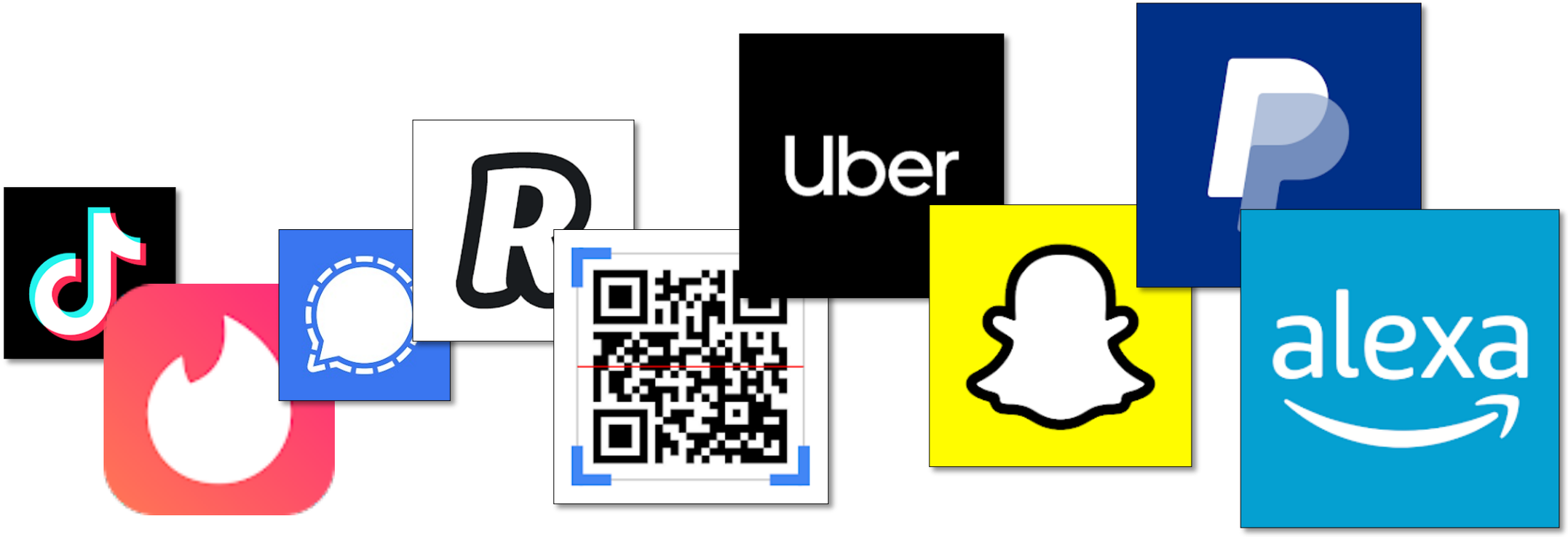
ARES 2023

Florian Draschbacher
Graz University of Technology, Graz, Austria
Secure Information Technology Austria, Vienna, Austria
florian.draschbacher@iaik.tugraz.at

August 31st, 2023

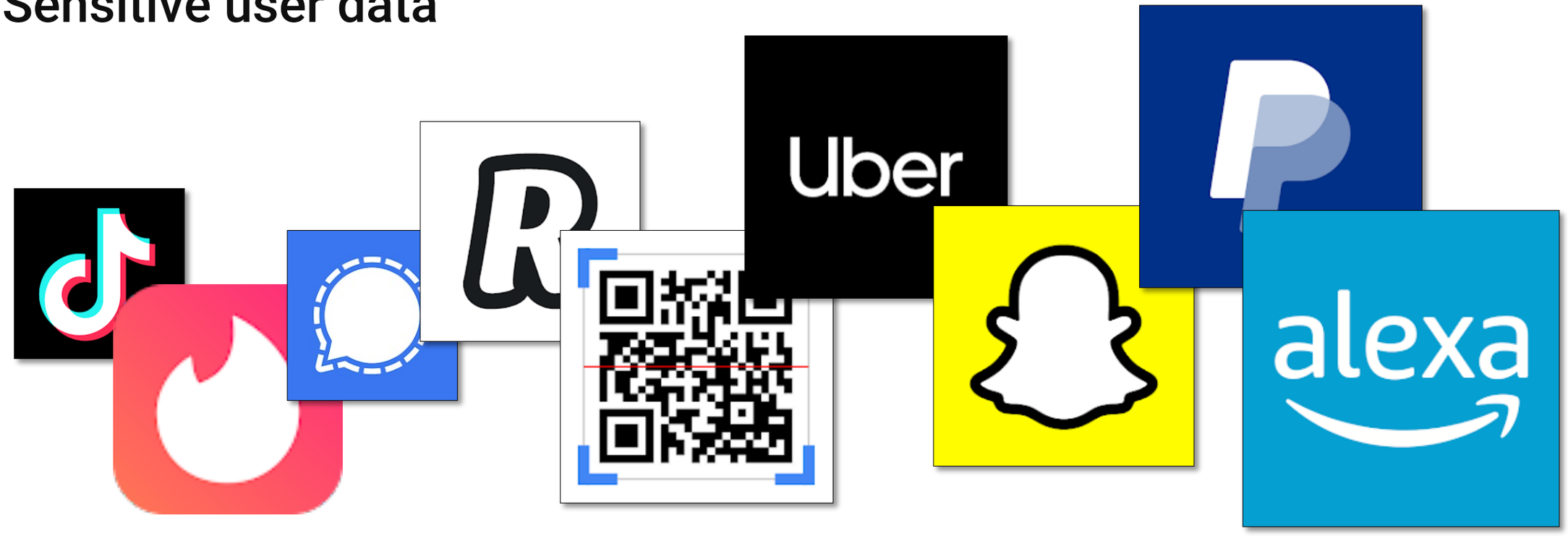
Mobile App Security

- New mobile computing use cases



Mobile App Security

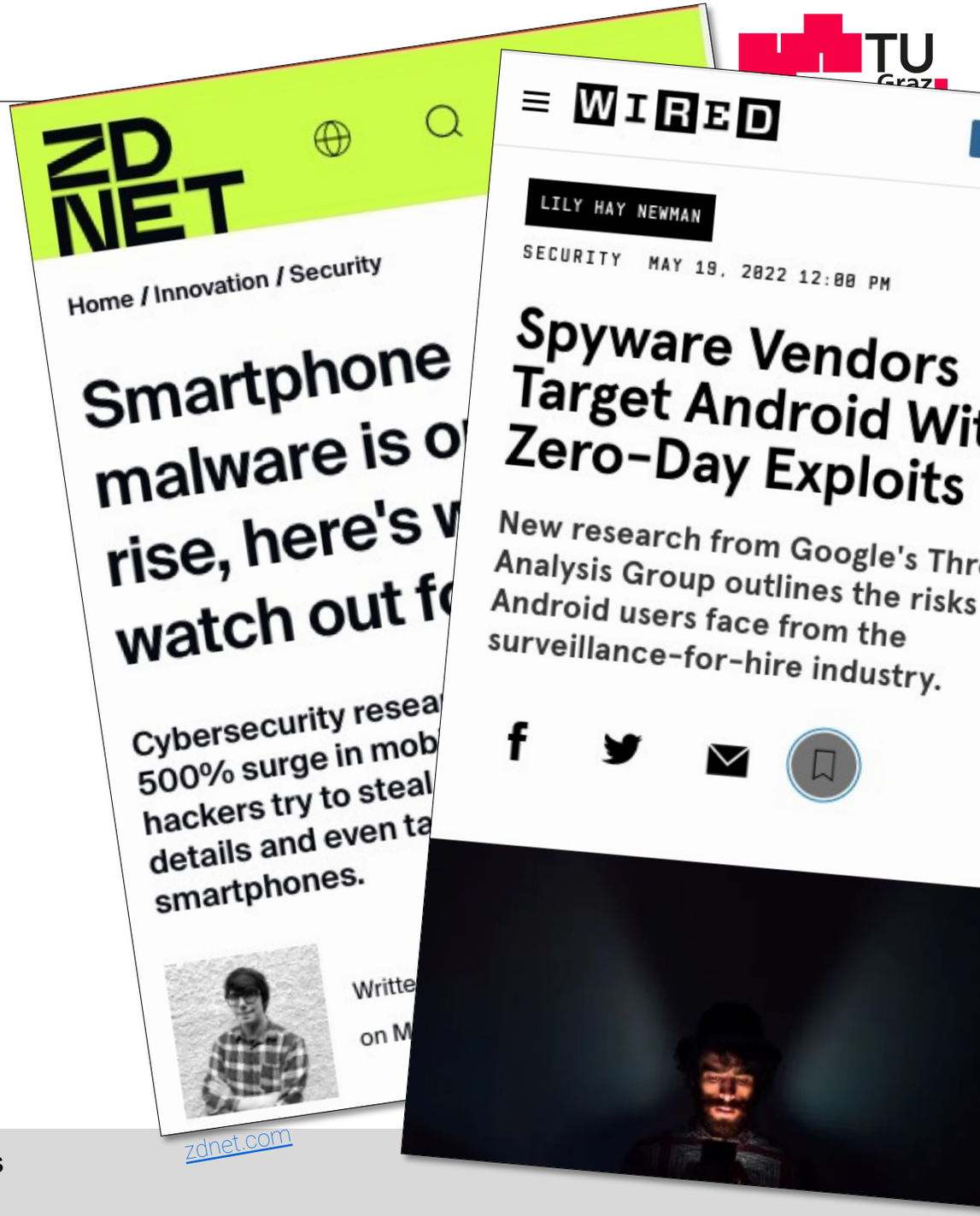
- New mobile computing use cases
  - Sensitive user data



Mobile App Security

- New mobile computing use cases
  - Sensitive user data

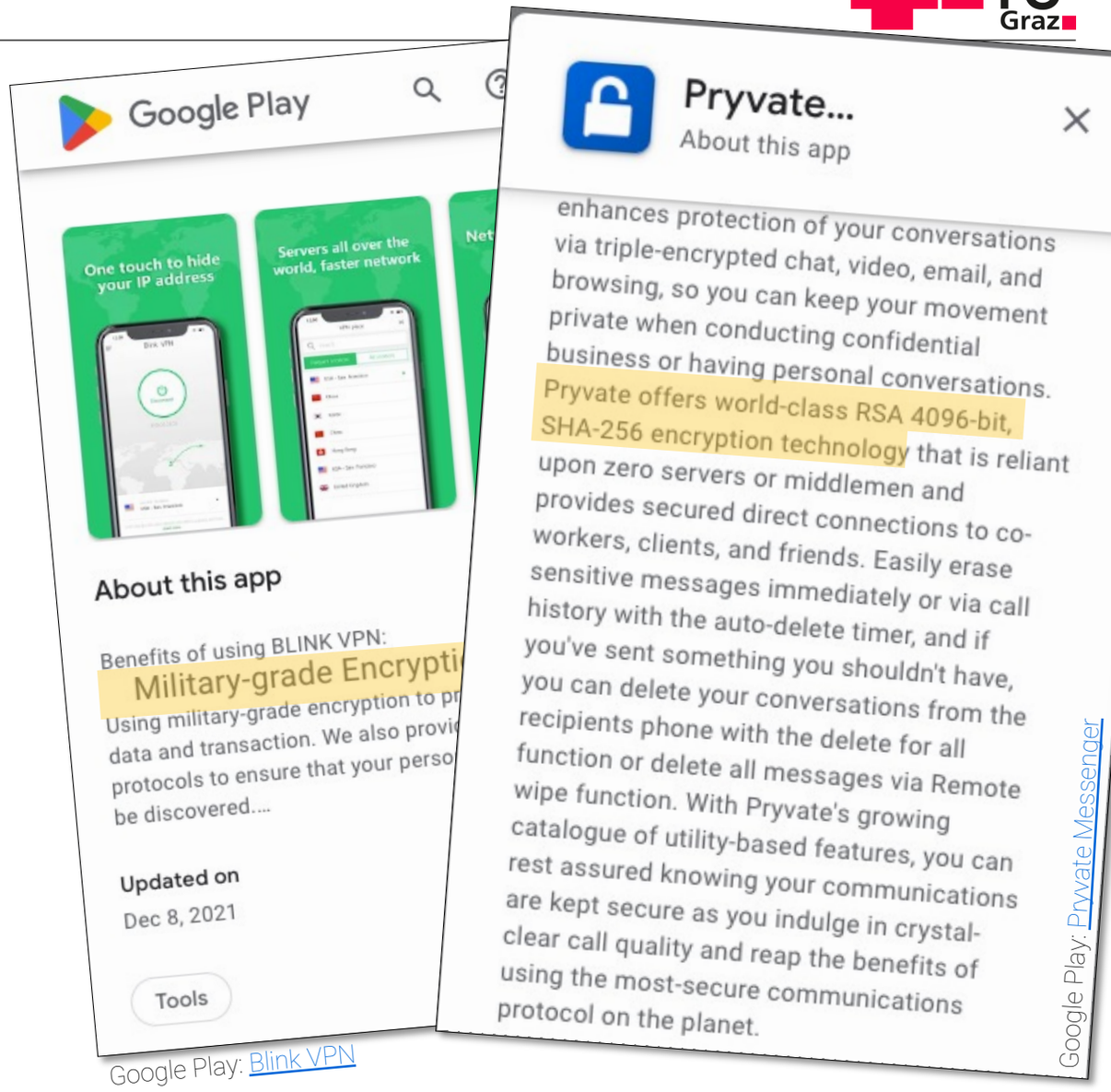
- Attractive for attackers



zdnet.com

Mobile App Security

- New mobile computing use cases
  - Sensitive user data
- Attractive for attackers
- Proper data protection?
  - Users rely on ecosystem



Google Play: Blink VPN

Google Play: Pryvate Messenger

Motivation

Motivation


KrebsonSecurity
In-depth security news and investigation

How Malicious Android Apps Slip Into Disguise

August 3, 2023

Researchers say mobile malware purveyors have been abusing a bug in the **Google Android** platform that lets them sneak malicious code into mobile apps and evade security scanning tools. Google says it has updated its app malware detection mechanisms in response to the new research.

Source: https://krebsonsecurity.com/2023/08/how-malicious-android-apps-slip-into-disguise/

Special Obfuscation Technique

Updates & Dynamic Code Loading

Malware on Google Play!


BLEEPINGCOMPUTER

Google explains how Android malware slips onto Google Play Store

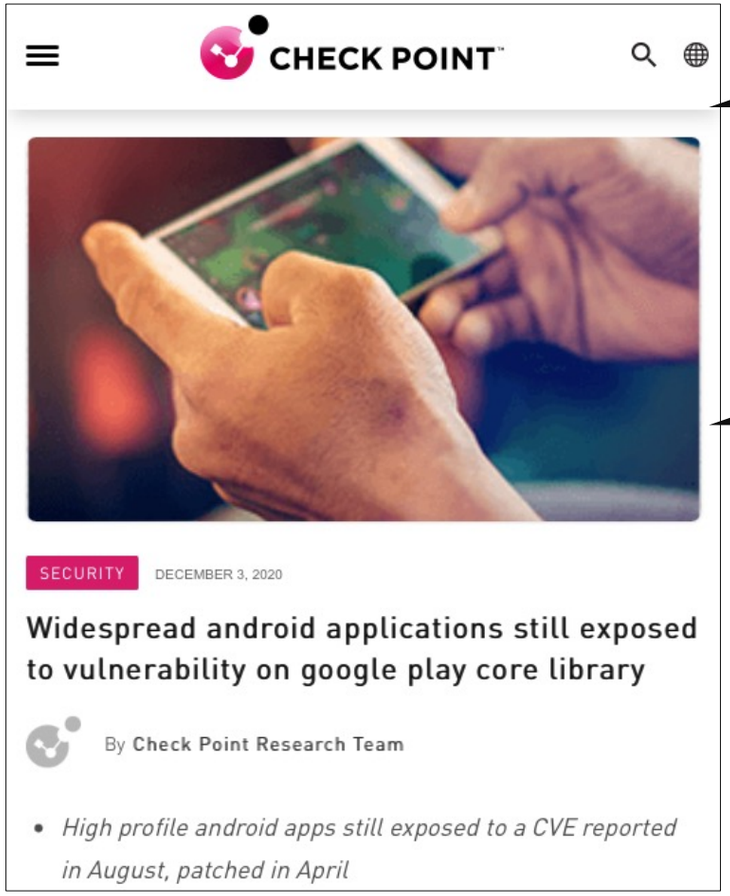
By Sergiu Gatlan

August 4, 2023 01:04 PM 0

The Google Cloud security team acknowledged a common tactic known as versioning used by malicious actors to slip malware on Android devices after evading the Google Play Store's review process and security controls.

Source: https://www.bleepingcomputer.com/news/security/google-explains-how-android-malware-slips-onto-google-play-store/

Motivation



CHECK POINT

SECURITY DECEMBER 3, 2020

Widespread android applications still exposed to vulnerability on google play core library

By Check Point Research Team

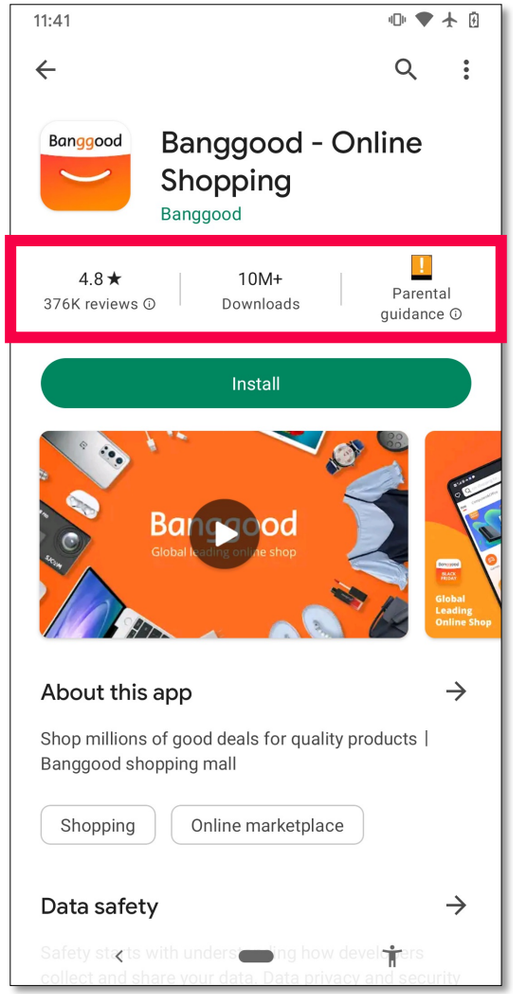
- *High profile android apps still exposed to a CVE reported in August, patched in April*

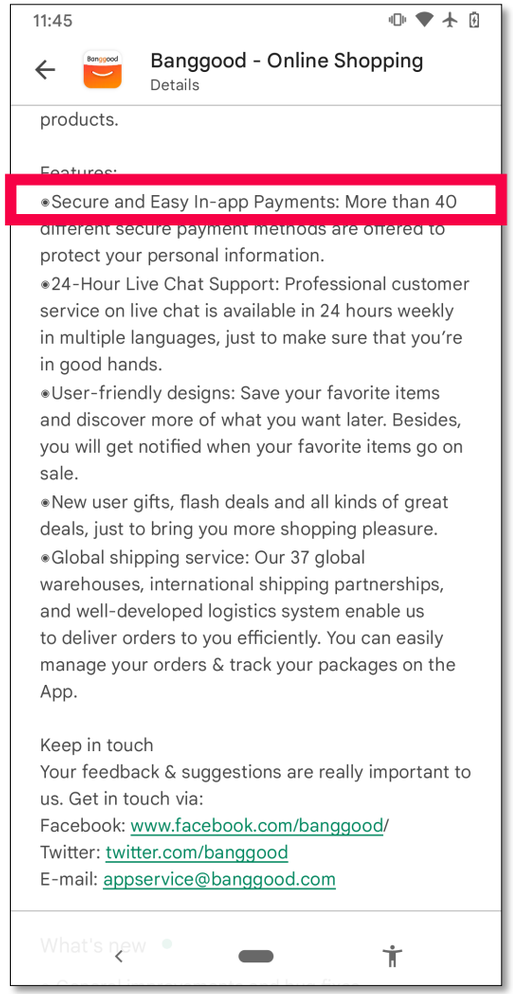
RCE due to vulnerable library

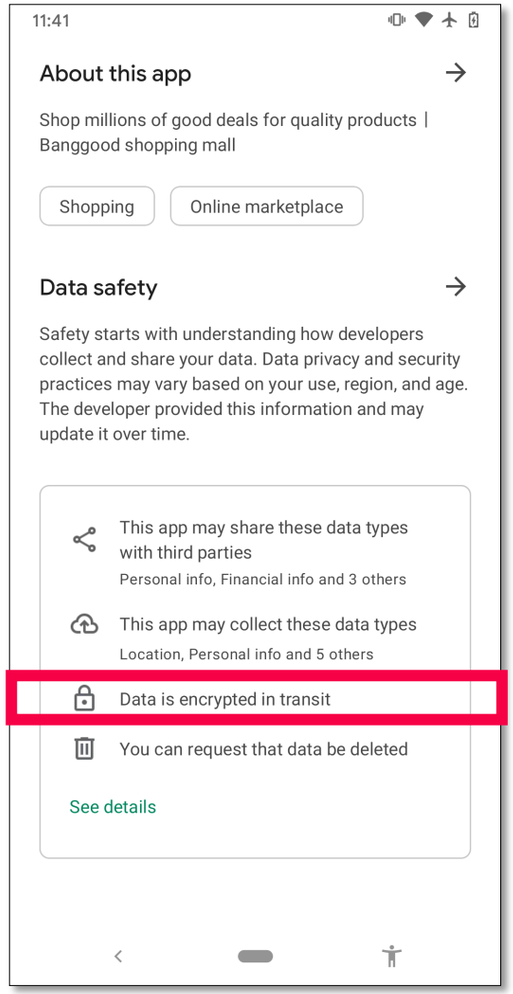
Months until apps used fixed library

Developers might not know about vulnerable library

Motivation

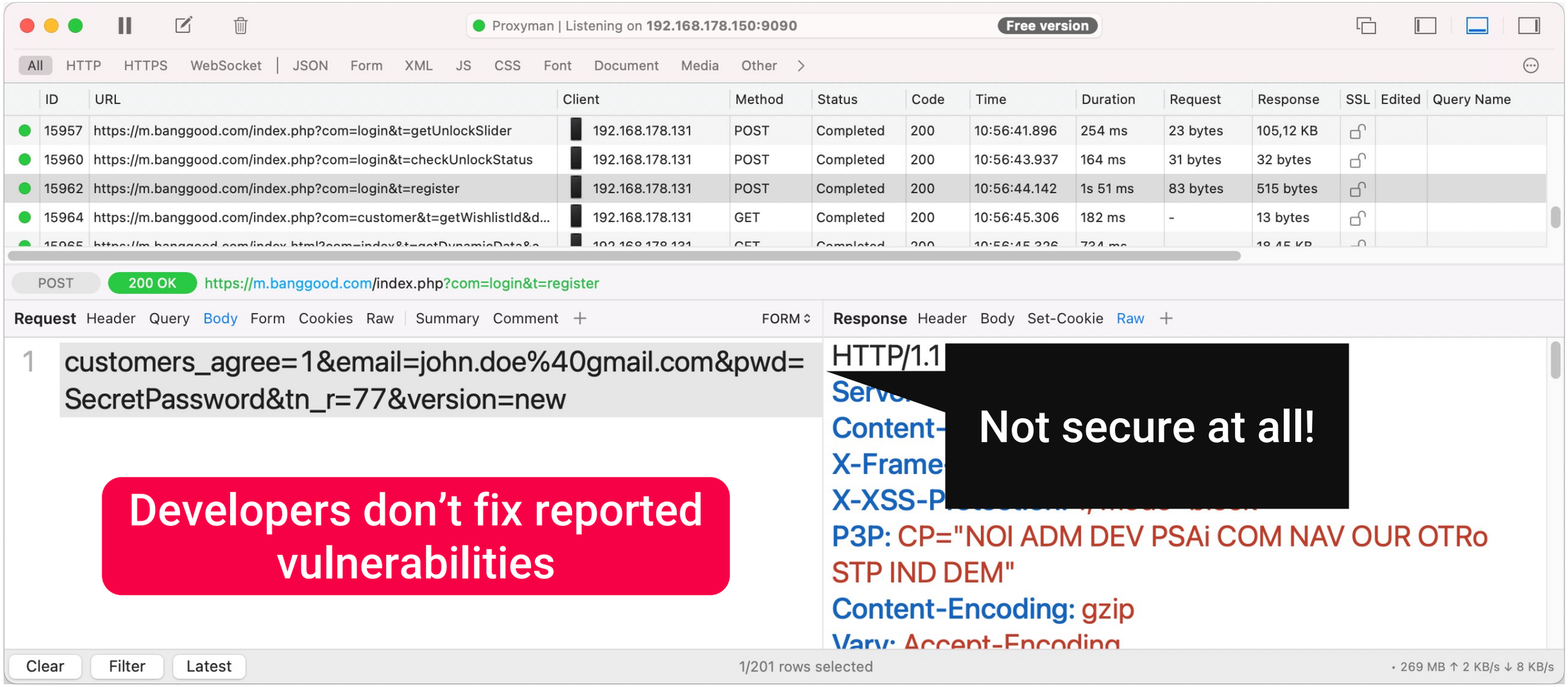






Apparently Secure?

Motivation


Proxyman | Listening on 192.168.178.150:9090
Free version
All HTTP HTTPS WebSocket JSON Form XML JS CSS Font Document Media Other
ID URL Client Method Status Code Time Duration Request Response SSL Edited Query Name
15957 https://m.banggood.com/index.php?com=login&t=getUnlockSlider 192.168.178.131 POST Completed 200 10:56:41.896 254 ms 23 bytes 105,12 KB
15960 https://m.banggood.com/index.php?com=login&t=checkUnlockStatus 192.168.178.131 POST Completed 200 10:56:43.937 164 ms 31 bytes 32 bytes
15962 https://m.banggood.com/index.php?com=login&t=register 192.168.178.131 POST Completed 200 10:56:44.142 1s 51 ms 83 bytes 515 bytes
15964 https://m.banggood.com/index.php?com=customer&t=getWishlistId&d... 192.168.178.131 GET Completed 200 10:56:45.306 182 ms - 13 bytes
POST 200 OK https://m.banggood.com/index.php?com=login&t=register
Request Header Query Body Form Cookies Raw Summary Comment FORM
1 customers_agree=1&email=john.doe%40gmail.com&pwd=SecretPassword&tn_r=77&version=new
Response Header Body Set-Cookie Raw
HTTP/1.1
Content-
X-Frame
X-XSS-P
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Content-Encoding: gzip
Vary: Accept-Encoding
Not secure at all!
Developers don’t fix reported vulnerabilities
Clear Filter Latest
1/201 rows selected
• 269 MB ↑ 2 KB/s ↓ 8 KB/s

Trusting the App Ecosystem…

- **Malware** finds way onto **Play Store**
  - Google‘s static checks not enough
- Developers **not aware** of threats
- Developers **fail to address** security issues
  - Innovation vs. Security
  - Same vulnerabilities reintroduced again and again

Trusting the App Ecosy…

- **Malware** finds way onto **Play Stor…**
  - Google‘s static checks not enoug…
- **Developers not aware** of threats [4]
- **Developers fail to address** security issues [6]
  - Innovation vs. Security
  - Same misuse reintroduced again and again

Research to the rescue!

Problem Statement & Approach

App Analysis and Vulnerability Mitigation

- Inspecting & manipulating execution flow of apps
  - Apply same set of changes to many apps

- No **app-agnostic holistic** tools available
  - Custom closed-source tools
  - Purpose-built code around app-specific tools

- **Research roadblock**

A2P2 – Android Application Patching Pipeline

Our Goal: An app-agnostic pipeline for manipulating Android apps
Use Cases: E.g. tracing API calls, mitigating vulnerabilities, …

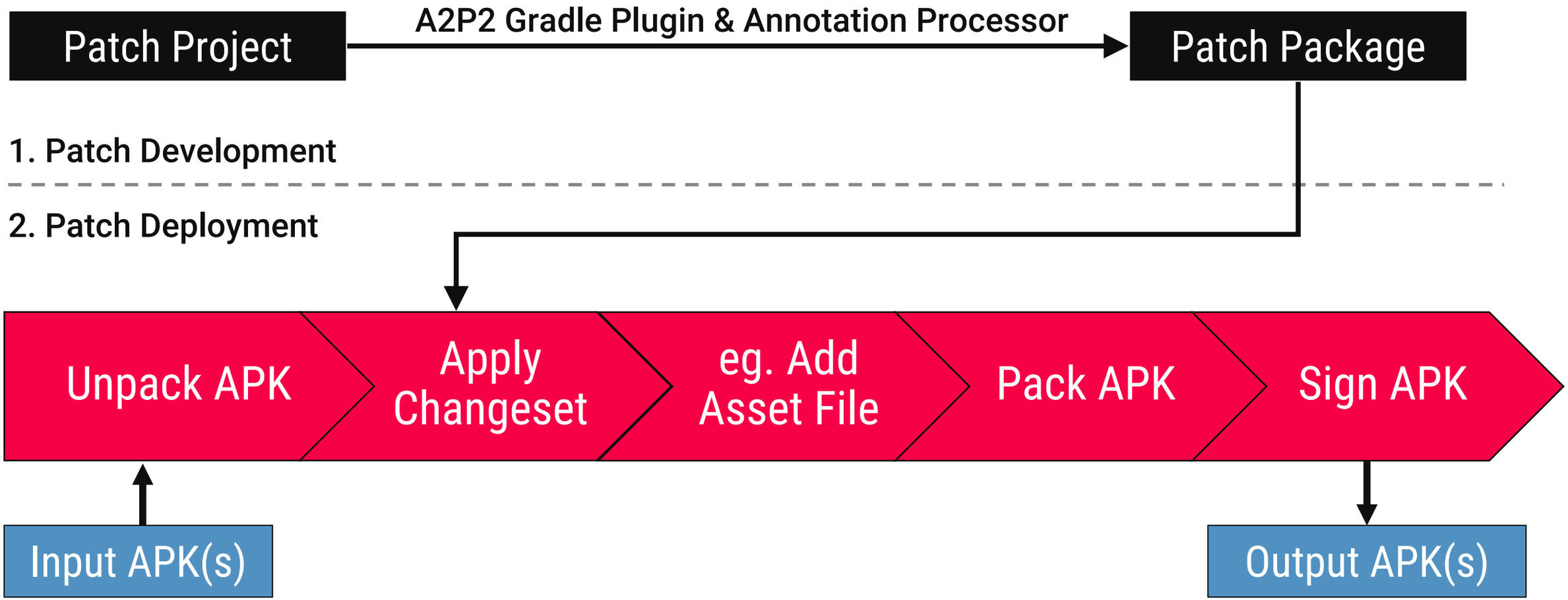
- **Easy to use**
  - Low entry barrier
- **Powerful**
  - High degree of customization
- **Open-Source**
  - Improve in a joint effort of the community

A2P2

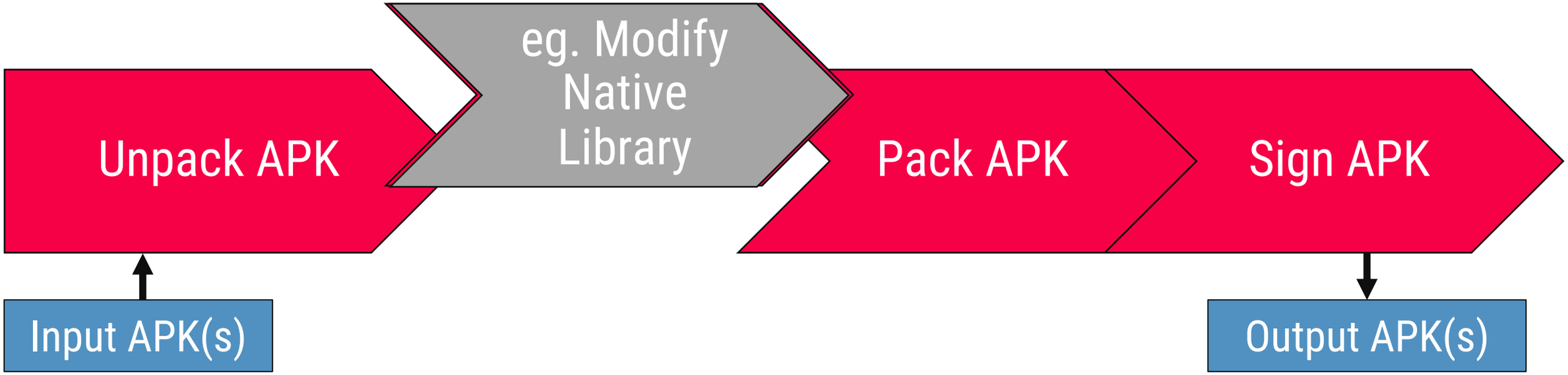
Two core concepts for achieving design goals

- **Custom Declarative Format** for application-agnostic changesets
  - Low entry barrier
- **Pipeline Architecture** with support for custom stages
  - Powerful customization

A2P2 – Basic Use Case



A2P2 – Advanced Use Cases



Standalone Pipeline Operation

A2P2

Patch projects **developed** in Android Studio IDE

- Annotation-based DSL for Java patches
- Custom patch format for manifest changes
- Support for merging resources, adding assets

A2P2 pipeline for applying patches or standalone operation

- Freely arrangeable parametrized stages
- Custom stages using low-level primitives

A2P2 – Example Declarative Patches

```java
public class TimeTravelPatch {
    @PatchClass({"java.lang.System"})
    @PatchStaticMethod
    public static long currentTimeMillis() {
        return OriginalMethods.java_lang_System.currentTimeMillis() - 1000*60*60;
    }
}
```

Java Patch

Manifest Patch

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:patch="http://schemas.android.com/apk/res-auto">

    <patch:add sel="manifest">
        <meta-data android:name="patched" android:value="true" />
    </patch:add>

    <patch:replace sel="manifest/@package">$${xpath(".")}.patched</patch:replace>
</manifest>
```

Evaluation

Performance

- **Minimal App Size Overhead**
  - 950 KB + Patch Size
- **Negligible Runtime Overhead**
  - Per-Call: < 44 ms
  - Per-Method: < 1.47 ms
- **High App Compatibility**
  - 91 % of 132 most popular Play Store apps

Example Patches

- **Detecting and mitigating crypto API misuse** (paper at AsiaCCS'23)
  - **Intercept crypto calls and upgrade arguments**
  - **Add `ContentProvider` for initialization**

- **Securing dynamic code loading**
  - **Intercept DEX loading and check integrity**
  - **Add asset file for configuration**

- **More: App cloning, injecting Flipper debugger, …**

Conclusion

A2P2

- First **application-agnostic wholistic patching** solution for Android apps
  - Readily usable for the research community
- Declarative patch format, pipeline architecture
  - Easy to use, yet powerful
- Example patches with focus on security

Questions?

Scan for full paper

Bibliography

[1]: Backes et al.: “AppGuard: Enforcing User Requirements on Android Apps”, *TACAS 2013*

[2]: Cao et al.: “Rotten Apples Spoil the Bunch: An Anatomy of Google Play Malware”, *ICSE 2022*

[3]: Draschbacher et al.: “CryptoShield – Automatic On-Device Mitigation for Crypto API Misuse in Android Applications”, *AsiaCCS 2023*

[4]: Duan et al.: “Identifying Open-Source License Violation and 1-day Security Risk at Large Scale”, *CCS 2017*

[5]: Falsina et al.: “Grab ‘n Run: Secure and Practical Dynamic Code Loading for Android Applications”, *ACSAC 2015*

[**6**]: Gao et al.: “Negative Results on Mining Crypto-API Usage Rules in Android Apps”, *MSR 2019*