TapTrap: Animation-Driven Tapjacking on Android

Philipp **Beer** | TU Wien
Marco **Squarcina** | TU Wien
Sebastian **Roth** | University of Bayreuth
Martina **Lindorfer** | TU Wien

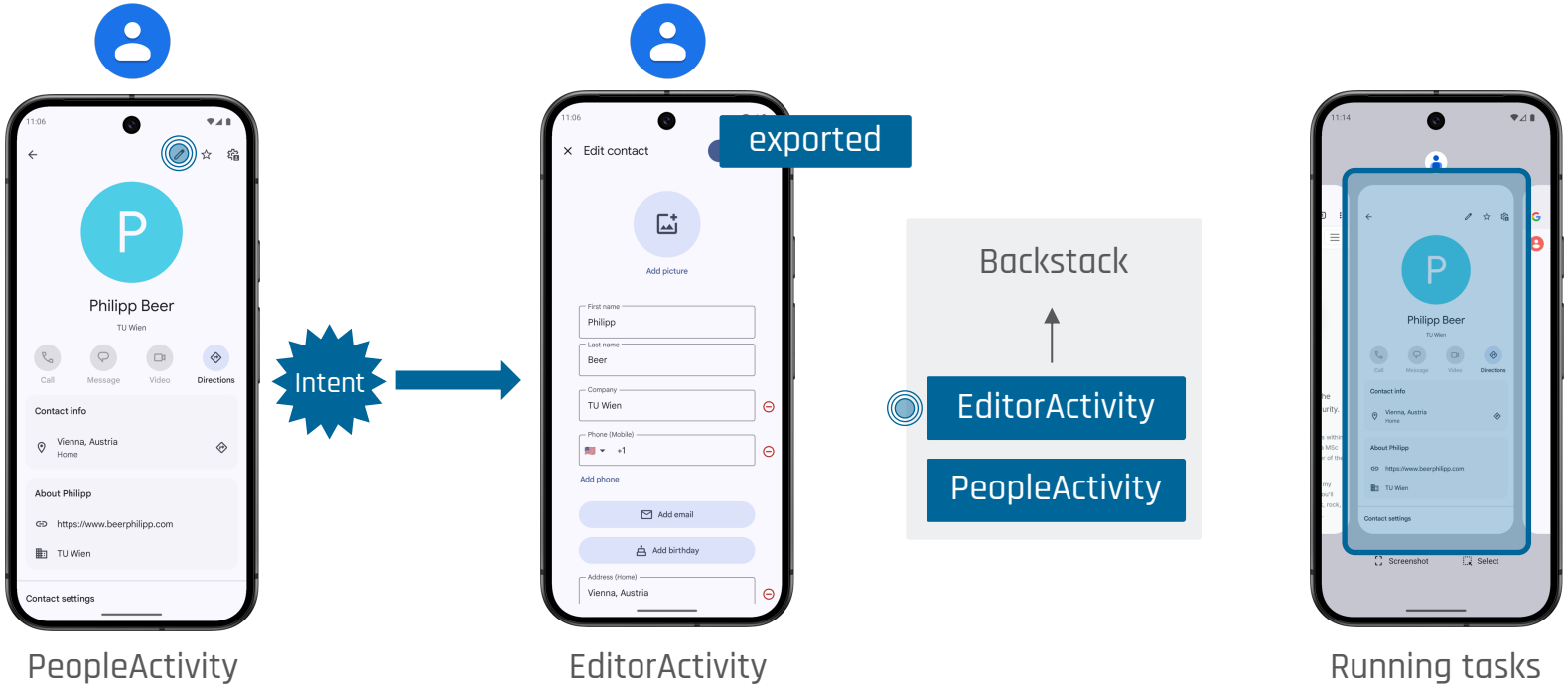
{philipp.beer, marco.squarcina}@tuwien.ac.at, sebastian.roth@uni-bayreuth.de, mlindorfer@iseclab.org

https://taptrap.click

34th USENIX Security Symposium 2025

Background

Background | Activity Fundamentals



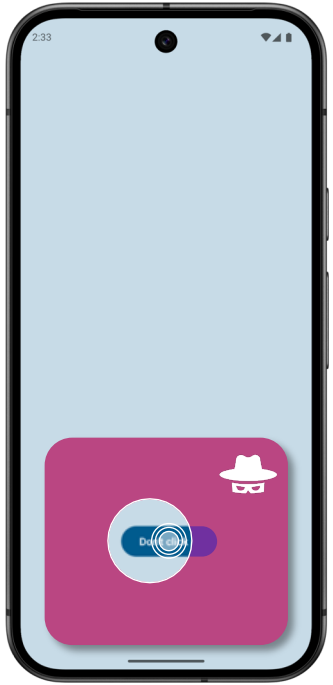
PeopleActivity EditorActivity Running tasks

Background | Previous Tapjacking Approaches

SAW* permission

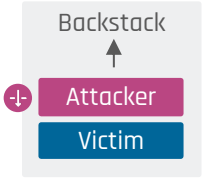
SAW notification

Mitigations assume the malicious app is **on top**



Detect overlays

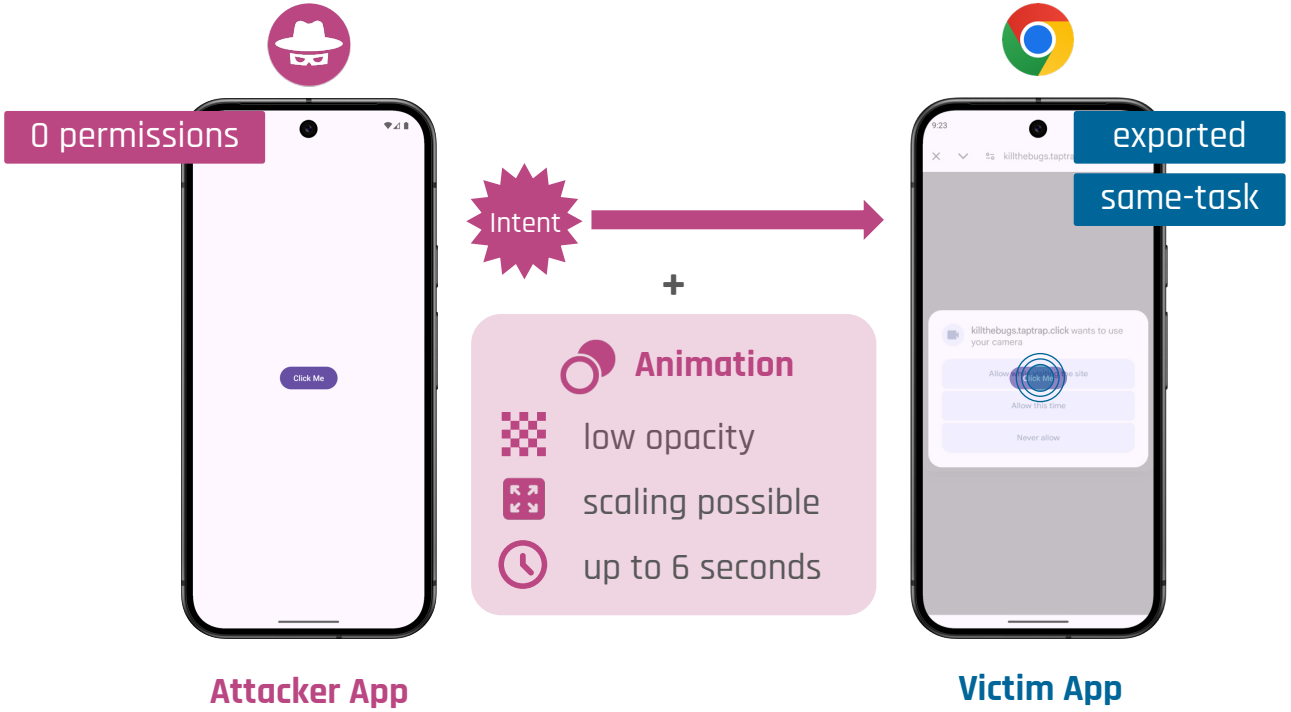
Block pass-through



*SAW = System Alert Window

TapTrap

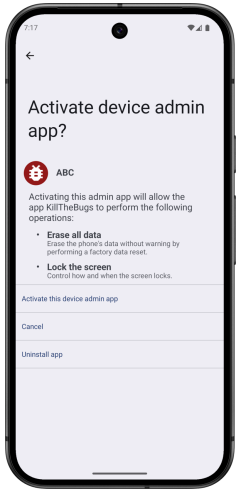
TapTrap | Threat Model & Mechanism



TapTrap | Implications (1/2)

Attack System Apps and Dialogs

Bypass runtime permissions
camera, microphone, location,
...



Device erasure
by requesting device admin permissions

and more

TapTrap | Implications (2/2)

Attack Browsers

Permission Bypass

Load **attacker-controlled website** in a Custom Tab that requests sensitive permission

Web Clickjacking

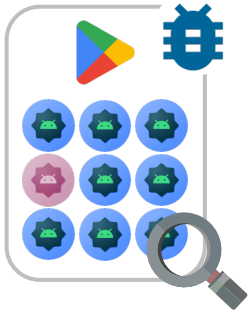
Open **victim website** in a Custom Tab and lure users into clicking sensitive button, e.g., “pay now”

Browser Vulnerability

- Chrome
- Samsung Internet
- ~~Opera~~
- ~~UC Browser~~
- Firefox (*)
- Edge
- Yandex
- Naver Whale
- Coc Coc
- Brave (*)

(*) requires 2 clicks to persist permissions

Attack 3rd Party Apps



App Vulnerability

App Vulnerability

Analysis of ~100K apps from the Google Play Store

An activity is vulnerable to TapTrap if it

is externally launchable

is same-task launchable

does not overwrite entry animations

does not wait for animation end before handling user input

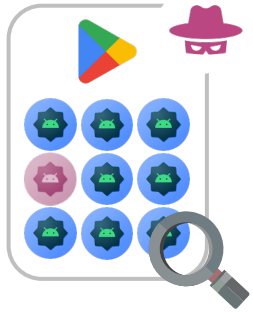
Manifest analysis

Bytecode analysis

76 **% of apps are vulnerable**

7 **% of activities are vulnerable**

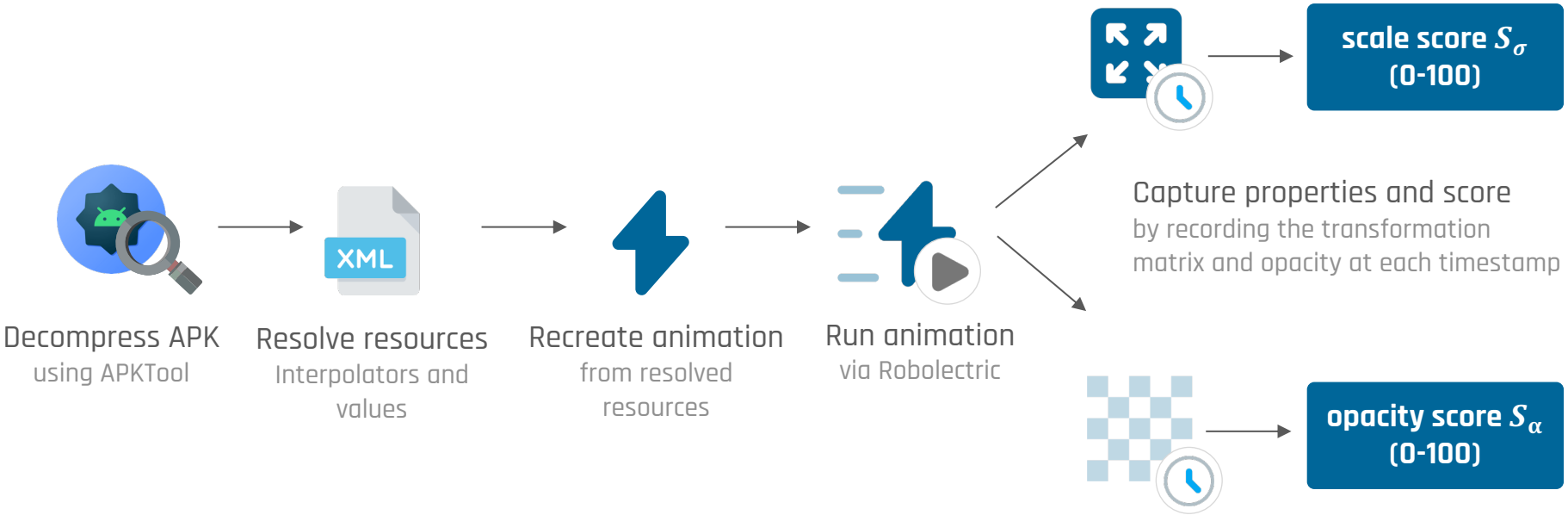
An app is vulnerable if it contains a vulnerable activity



In The Wild

Exploitation In The Wild | Methodology

Analysis of ~100K apps from the Google Play Store



Decompress APK
using APKTool

Resolve resources
Interpolators and values

Recreate animation
from resolved resources

Run animation
via Robolectric

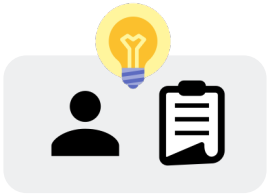
Exploitation In The Wild | Results

Finding #1: Animations can exceed the predefined limit

Our pipeline identified that animations can last longer than 3 seconds (up to 6 seconds)

Finding #2: TapTrap is not exploited in the wild

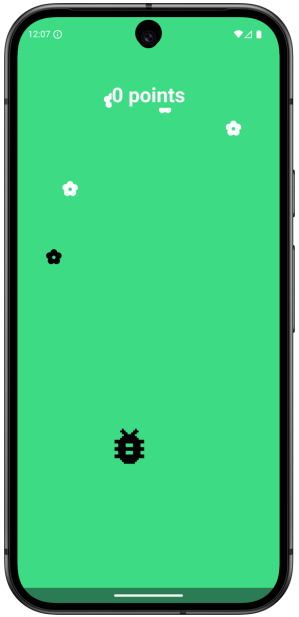
While 28 apps had $S_\sigma \geq 50 \vee S_\alpha \geq 50$, manual analysis of these showed no evidence of exploitation



User Awareness

User Awareness

User study with 2 rounds to evaluate user awareness



User Study App “KillTheBugs”

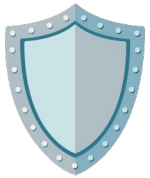
Level 1
Request location permission via Chrome Custom Tab

Level 2
Request camera permission via Chrome Custom Tab

Level 3
Trick user into granting the device administrator permission

20 **participants**
recruited from entry-level bachelor courses and via word of mouth

all **failed to detect at least one TapTrap variant**
even after being informed that the app is malicious



Mitigations

Mitigations

Android

ignore touches under a certain opacity threshold or over a scale threshold

Android 16 still vulnerable*

GrapheneOS fixed

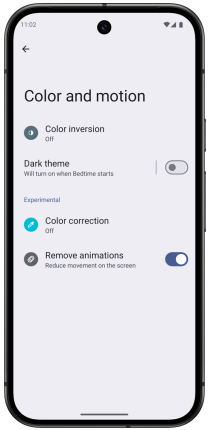
Apps

override entry animation

wait for animation to finish before handling user input

fixed

Users



disable animations

* as of July 23, 2025

In the paper

Proposing TapTrap

Animation-based tapjacking attack

Classification of Previous Tapjacking Attacks/Mitigations

First to exploit activity transition animations

Exploitation in the Wild

Analysis of ~100K apps finding no evidence of exploitation

App Vulnerability

Analysis of ~100K apps finding 76% are vulnerable to TapTrap

User Awareness

User study with 20 participants finding that all failed to notice at least one TapTrap variant

demonstration video, artifacts, and more info

https://taptrap.click

This work was supported by the Vienna Science and Technology Fund (WWTF) and the City of Vienna, the Austrian Science Fund (FWF), and SBA Research.

For image attribution, see next slide

Image Attribution

The presentation includes the following images and icons:

Trademarked Logos

Google Icons (Apache 2.0)

lucca fruzza (from Noun Project, CC BY) https://thenounproject.com/icon/through-1595346/

Soco St (CC Attribution) https://www.svgrepo.com/svg/493162/hacker

Blivesta (MIT) https://www.svgrepo.com/svg/506667/person

Flat Icon Design (PD) https://www.svgrepo.com/svg/485268/magnifying-glass

Icooon Moon (PD) https://www.svgrepo.com/svg/479733/questionnaire https://www.svgrepo.com/svg/483652/hacker

Googlefonts (Apache) https://www.svgrepo.com/svg/398289/shield

Iconsax (MIT) https://www.svgrepo.com/svg/496194/flash-1

Recap Kütük (CC Attribution) https://www.svgrepo.com/svg/426798/clock

Radix UI (MIT) https://www.svgrepo.com/svg/361642/transparency-grid

Solar Icons (CC Attribution) https://www.svgrepo.com/svg/526106/play

SVG Repo (CC0) https://www.svgrepo.com/svg/474868/idea https://www.svgrepo.com/svg/255827/xml

Pancaokta (CC Attribution) https://www.svgrepo.com/svg/382177/book-education-idea-learning-school-study

Other Web images https://developer.android.com/develop/ui/views/launch/icon_design_adaptive

GrapheneOS https://github.com/GrapheneOS/branding-extra/blob/main/simple_with_text.svg

https://commons.wikimedia.org/wiki/File:Google_Contacts_logo.png