Friedrich-Alexander-Universität
Erlangen-Nürnberg

The Evolution of Trusted UI on Mobile

A Systematization of Knowledge

Davide Bove

June 1, 2022

IT Security Infrastructures Labs
Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Table of contents

Contributions

- Overview of UI-related issues on Android (main attack vectors, design issues)
- Evaluation of current research (2014-2020) and current device protections against UI attacks
- Classification of systemic weaknesses and future research directions

Background

Android UI

- Android is based on Linux
- Apps have unique UIDs
- Apps are sandboxed:
  - Linux process isolation
  - File system permissions (rwx etc)

  ⇒ An app can not access data or processes of other apps (same for UI)

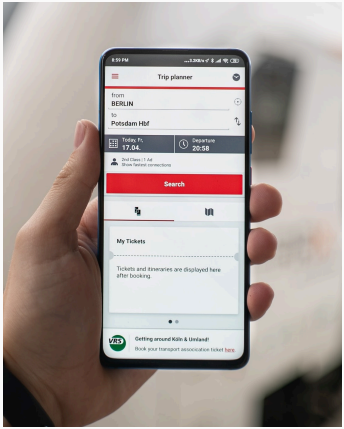


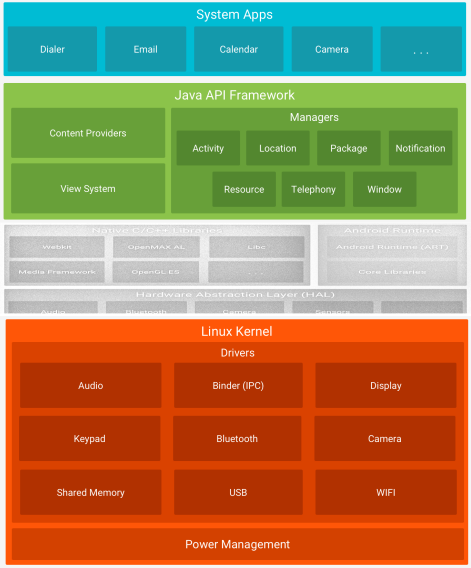
Photo by Mika Baumeister on Unsplash

Android UI

No interaction? Booooooring!

Solution: Inter Process Communication

- Window Manager
- Activity Manager
- View System
- Binder



ARM TrustZone

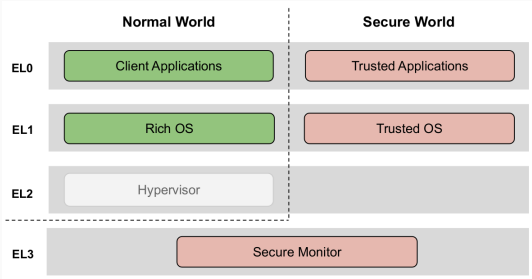
Trusted Execution Environment (TEE)

Goal: give security guarantees for specific applications

- Even if OS compromised
- Even if hardware compromised

Main mechanisms:

- Hardware isolation of software
- Encryption



TrustZone Architecture

Issues

Issues & Defenses

- **I01.** Missing indicators
- **I02.** Unprivileged access to overlays
- **I03.** Overlays covering information
- **I04.** Apps can hijack the window stack
- **I05.** Lack of alternatives
- **I06.** Highly privileged system access

- **D01.** Touch Filtering
- **D02.** Limiting overlay priority
- **D03.** Additional indicators
- **D04.** Hiding overlays for critical dialogs
- **D05.** Secure system dialogs
- **D06.** Overlay detection
- **D07.** Clickjacking detection
- **D08.** App hijacking detection
- **D09.** UI Sandboxing
- **D10.** UI as a trusted app
- **D11.** Single trusted UI components
- **D12.** Dedicated LED indicator
- **D13.** Physical separation

Issues Classification

Overlays / Context Hiding
clickjacking, DoS, deception

I01. Missing indicators
I02. Unprivileged access to overlays
I03. Overlays covering information
I04. Apps can hijack the window stack

UI control
(full) takeover, privacy leak

I05. Lack of alternatives
I06. Highly privileged system access

Issues Classification

Overlays / Context Hiding

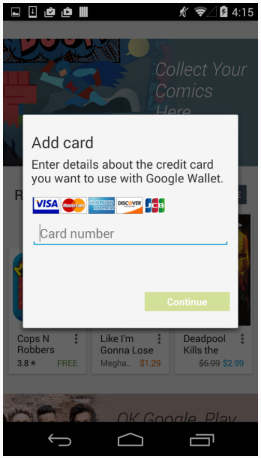
clickjacking, DoS, deception

I01. Missing indicators

I02. Unprivileged access to overlays

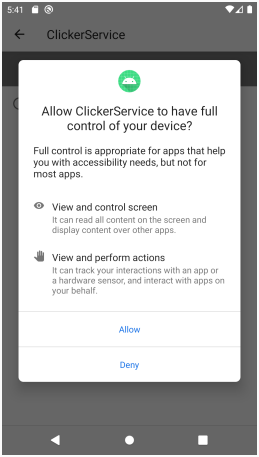
I03. Overlays covering information

I04. Apps can hijack the window stack



Banking Trojan “Acecard” [Source]

Issues Classification



Example of accessibility service

UI control

(full) takeover, privacy leak

I05. Lack of alternatives

I06. Highly privileged system access

Defenses

Defenses

Implemented in Android

- D01. Touch Filtering
- D02. Limiting overlay priority
- D03. Additional indicators
- D04. Hiding overlays for critical dialogs
- D05. Secure system dialogs

Kernel and OS-based

- D06. Overlay detection
- D07. Clickjacking detection
- D08. App hijacking detection
- D09. UI Sandboxing

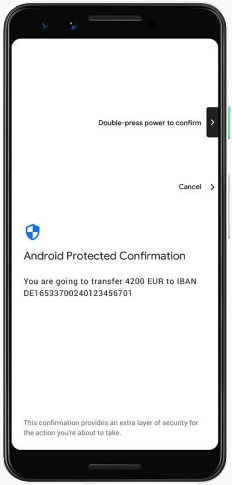
TEE-based

- D10. UI as a trusted app
- D11. Single trusted UI components
- D12. Dedicated LED indicator
- (D13. Physical separation)

Secure system dialogs

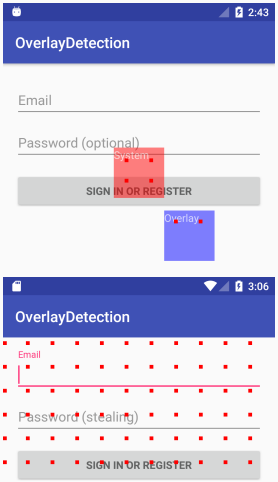
Protected Confirmation

- Hardware-protected user interface
- Two parts residing in TEE
  - **Keymaster**: for generating keys
  - **ConfirmationUI**: generates cryptographic statement



Source: AOSP (CC BY 4.0)

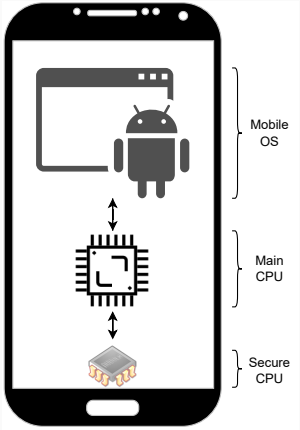
Overlay detection



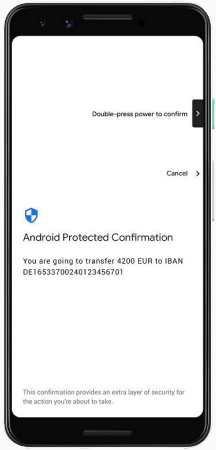
Window Punching

- App hardening measure
- Used to detect overlays in combination with Touch Filtering
- App manually simulates touches on the screen to detect overlay.

Physical separation

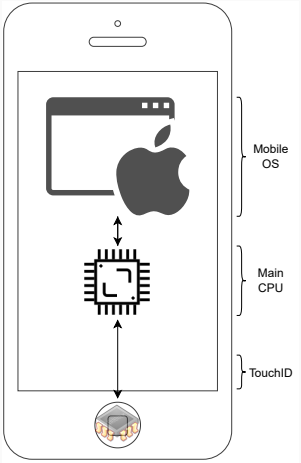


Android

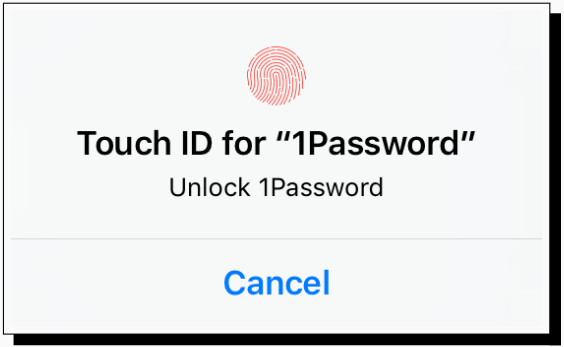


Protected Confirmation uses Titan-M

Physical separation



Apple iPhone



TouchID prompt [Source]

Conclusion

Summary

	Overlays / Context Hiding	UI control
Description	clickjacking, DoS, deception	(full) takeover, privacy leak
Issue(s)	I01 – I04	I05 – I06
Defenses	D01 – D04, D06 – D08	D05, D09 – D11
Threat model	USR	OS

Overview of issues in research, suggested defenses and assumed threat model

Summary

Shortcomings:

- Almost no consideration for the end user
- Shift from pure OS-level measures to HW-supported and TEE-based
- Shift to co-processors does not improve security by itself

Thanks for watching

Paper is Open Access

Download this presentation: https://d4vi.de/sok-trusted-ui-presentation

Download the paper: https://d4vi.de/sok-trusted-ui