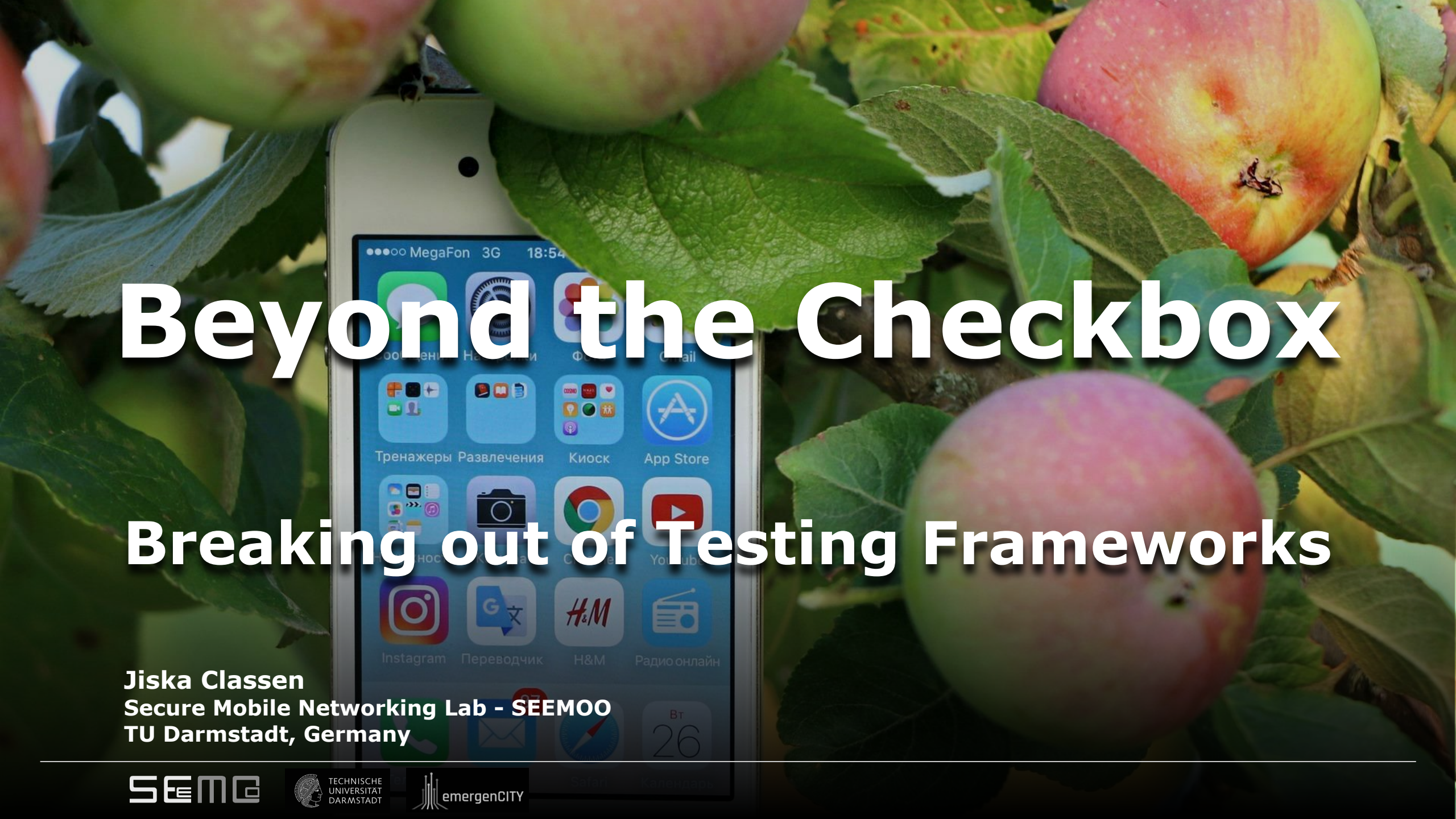


Beyond the Checkbox

Breaking out of Testing Frameworks

Jiska Classen
Secure Mobile Networking Lab - SEEMOO
TU Darmstadt, Germany

SEEMOO TECHNISCHE UNIVERSITÄT DARMSTADT emergenCITY

OWASP MASVS

GitHub Repo

The **OWASP MASVS (Mobile Application Security Verification Standard)** is the industry standard for mobile app security. It can be used by mobile software architects and developers seeking to develop secure mobile applications, as well as security testers to ensure completeness and consistency of test results.

Download the MASVS v2.0.0

Starting with MASVS v2.0.0, translations will no longer be included to focus on the development of MASTG v2.0.0. We encourage the community to create and maintain their own translations. Thank you to all the past translators who generously volunteered their time and expertise to make the MASVS accessible to non-English speaking communities. We truly appreciate your contributions and hope to continue working together in the future. The past MASVS v1 translations are still available in the MASVS repo.

MASVS v1.5.0

Security Checklist for Complete, Consistent Results

✅ Obfuscate app as resilience measure
✅ TLS communication
✅ No sensitive data stored locally
✅ …

Yay, we’re compliant!

Real-World Attack Complexity

😎 Real-World Attackers

“I let them connect to my free Starbucks Wi-Fi and then…”

“Phishing didn’t work, but we still have that iMessage 0-day exploit just in case.”

Sophisticated attacks happen, and this is how actual 0-days harming users work.

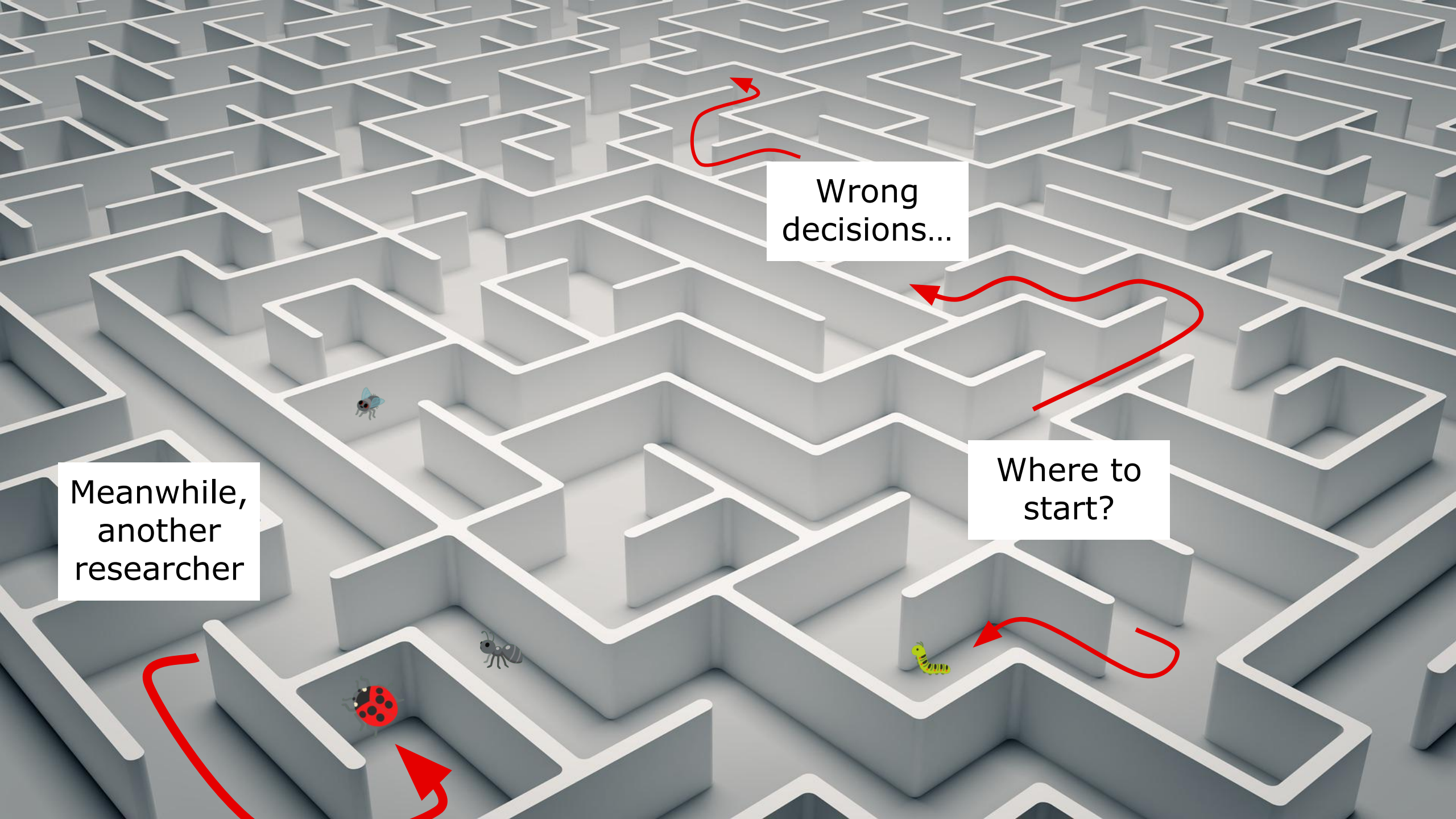
We don’t need testing guides.

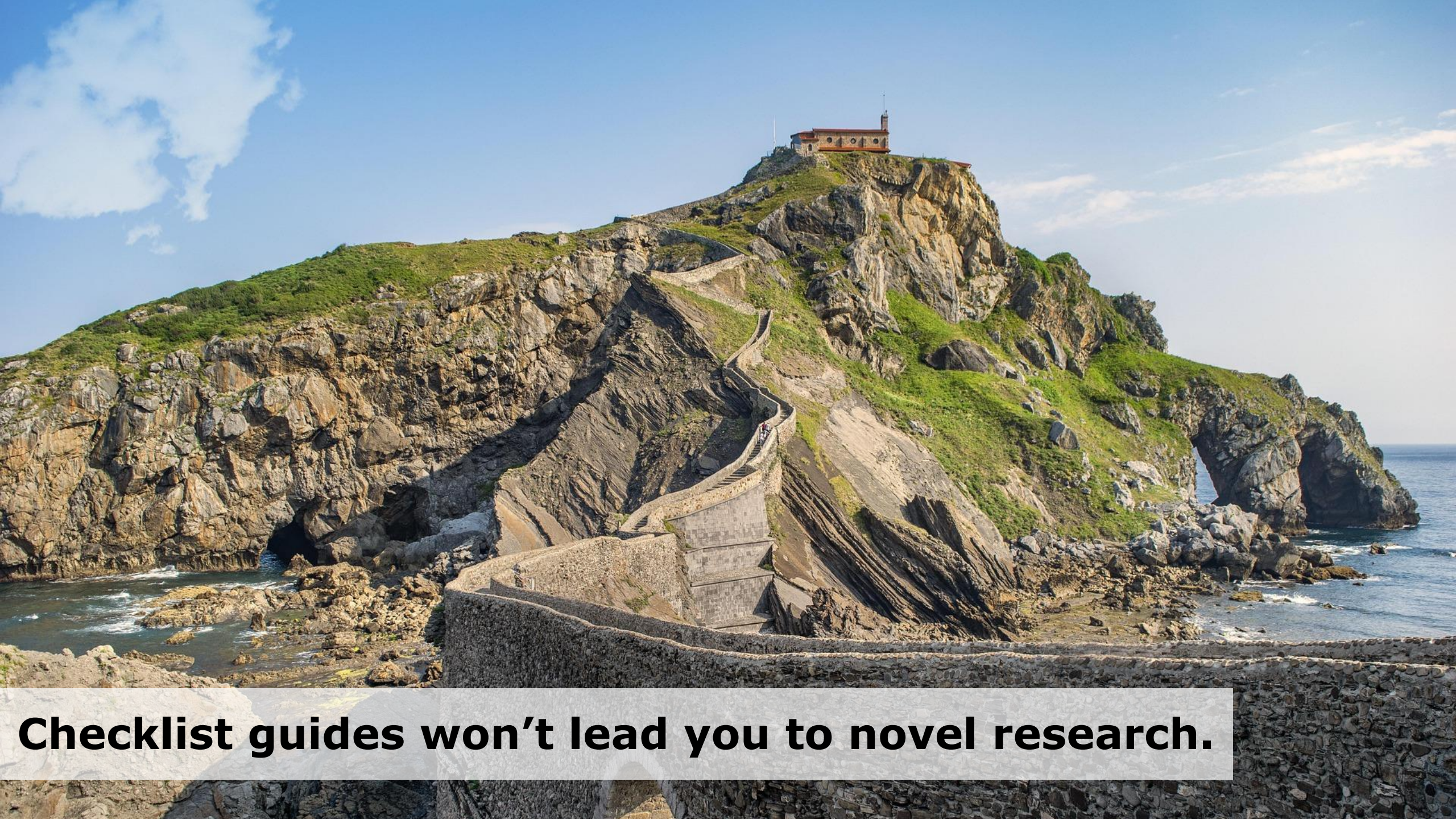
Security Research in Academia

- Pick an interesting target nobody analyzed before.
- Find a new bug class.
- Research a new bug finding method.
- …

} Novelty

There’s no bug discovery guarantee.

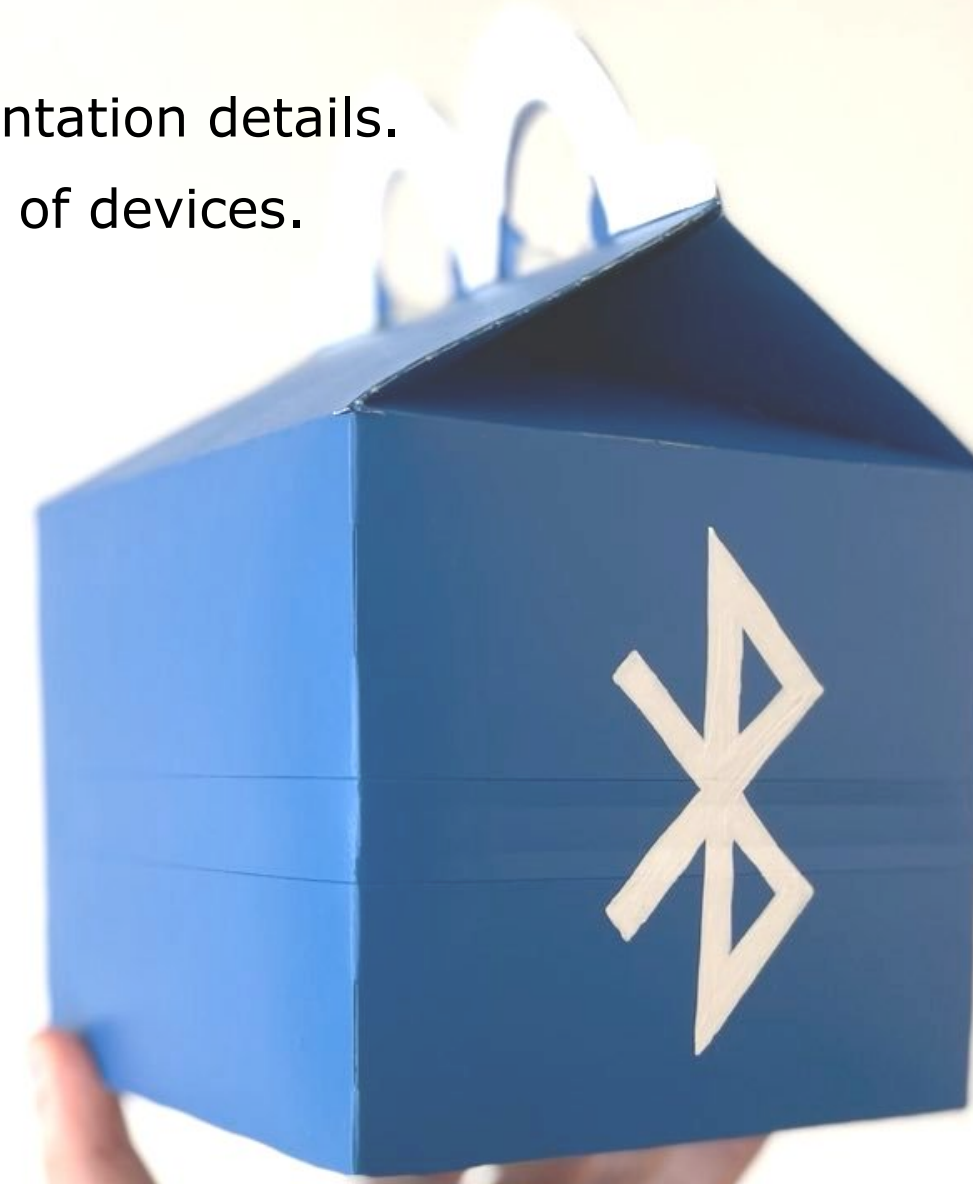

Wrong decisions…
Where to start?
Meanwhile, another researcher



Checklist guides won’t lead you to novel research.

My Journey: Bluetooth Security Research

- No affordable tooling to test cryptographic implementation details.
  Severe bugs existed in the specification and billions of devices.



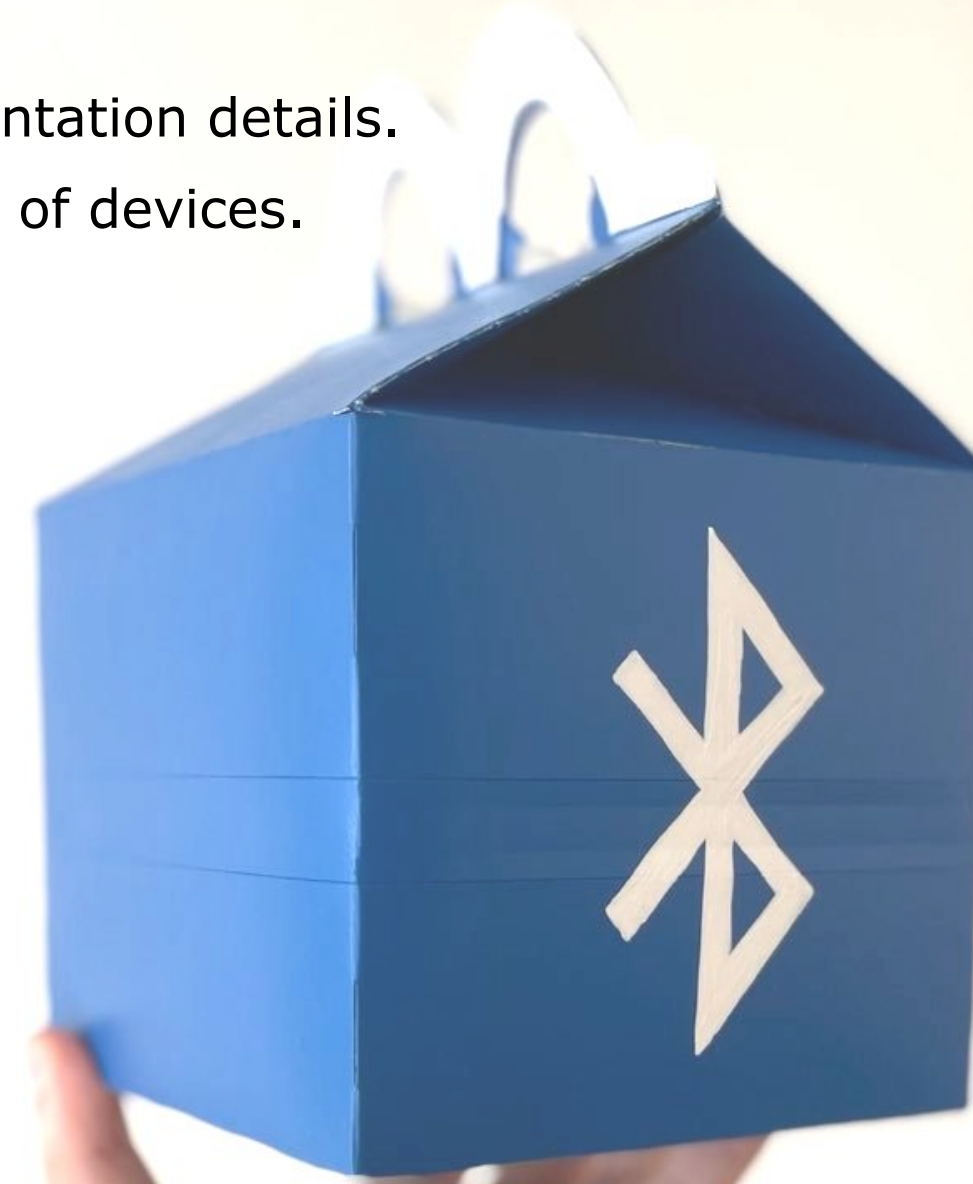
My Journey: Bluetooth Security Research

- No affordable tooling to test cryptographic implementation details.

  Severe bugs existed in the specification and billions of devices.

- Missing understanding of wireless attack surfaces.

  Inter-chip attacks for Bluetooth to Wi-Fi escalation, iPhone Bluetooth malware after power off.



My Journey: Bluetooth Security Research

- No affordable tooling to test cryptographic implementation details.

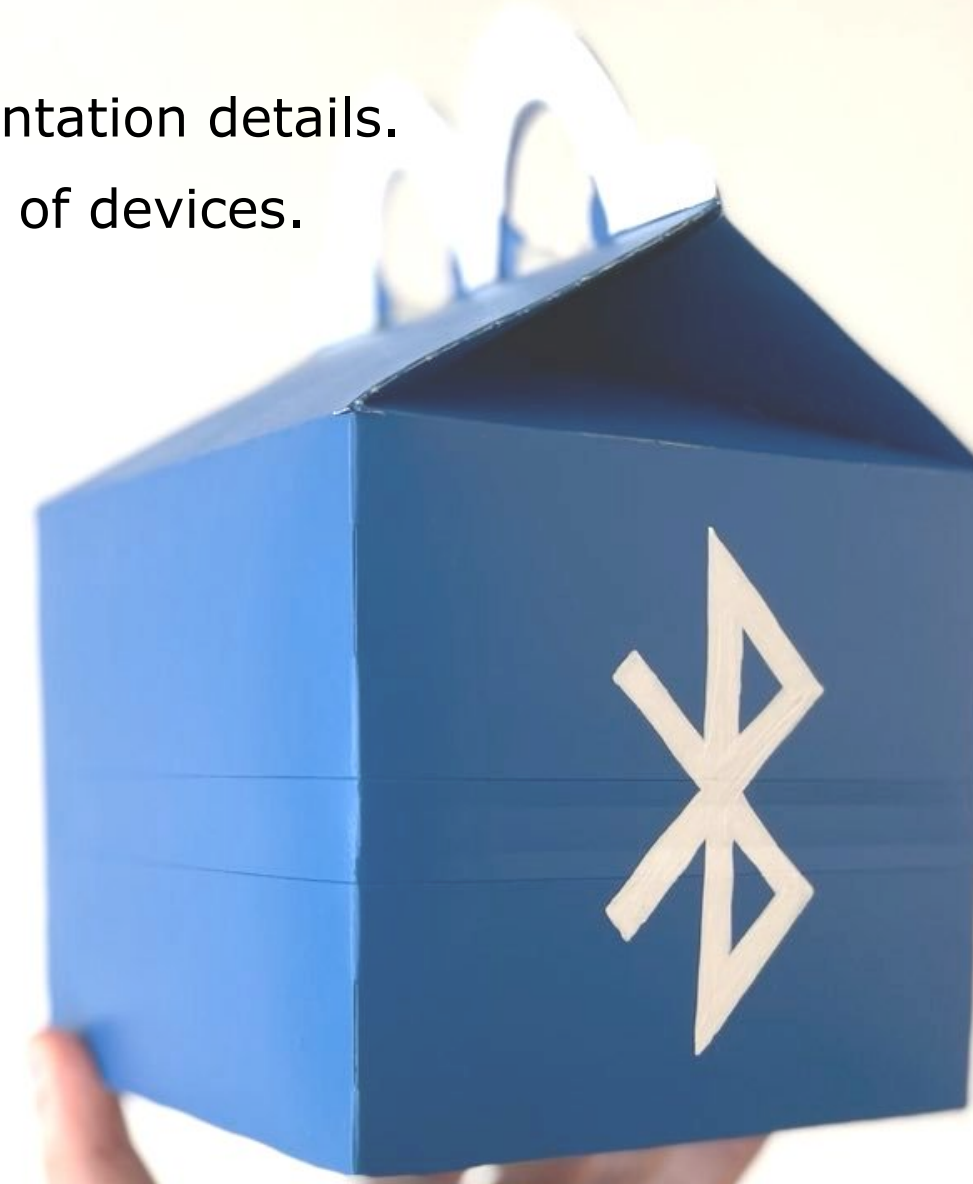
  Severe bugs existed in the specification and billions of devices.

- Missing understanding of wireless attack surfaces.

  Inter-chip attacks for Bluetooth to Wi-Fi escalation, iPhone Bluetooth malware after power off.

- Existing fuzzers incapable of Bluetooth stacks.

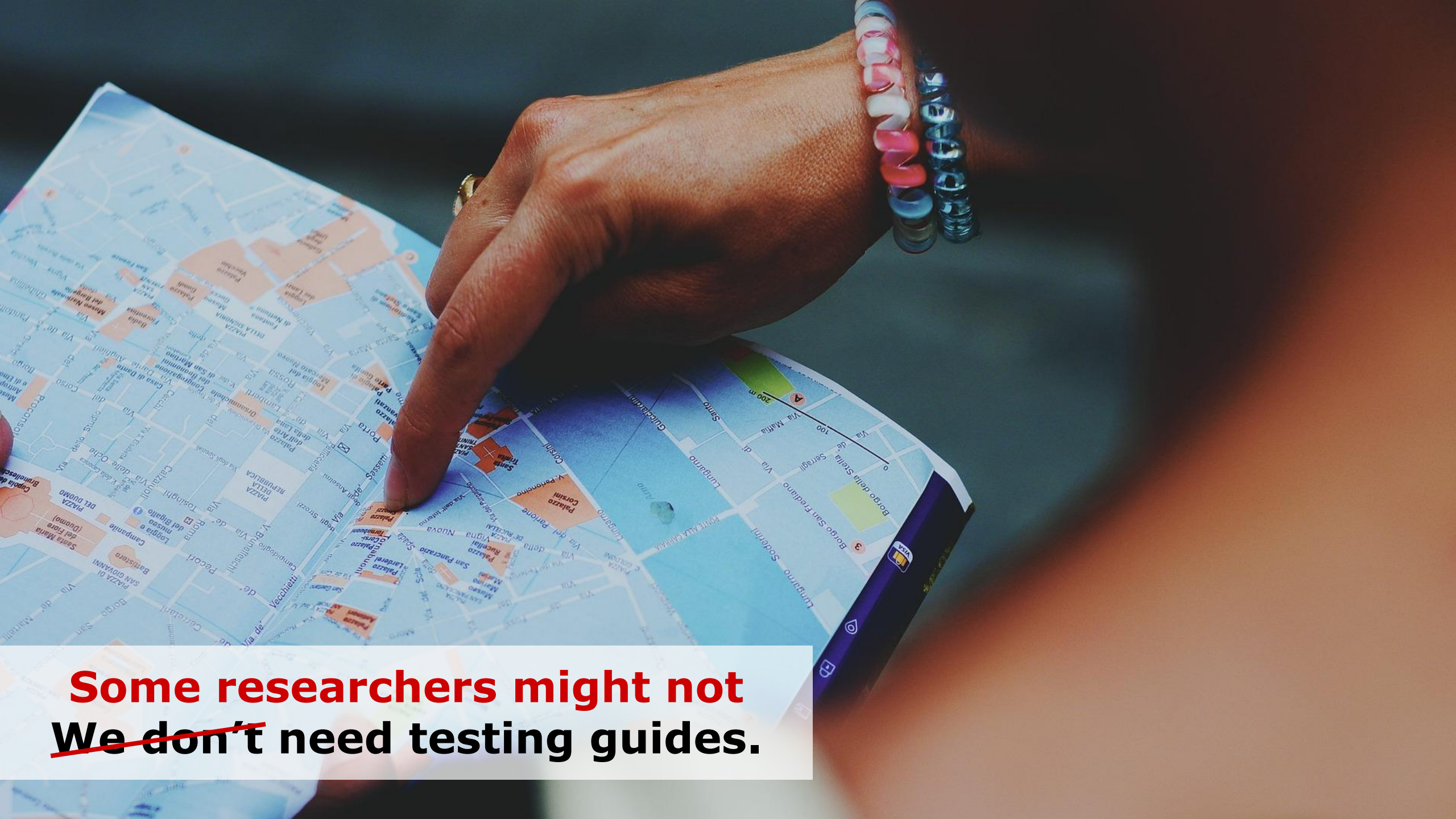
  Novel fuzzers for Bluetooth firmware & iOS, significant extensions for Linux.



Where I started, there were no testing guides, just unanswered questions.

Pentesting & CTF as Thesis

- Many students knew they would like to become pentesters.
- Mostly interested into web, mobile applications, and IoT.
- How to systematically test these?



Some researchers might not

~~We don't~~ need testing guides.

Pentesting & CTF as Thesis

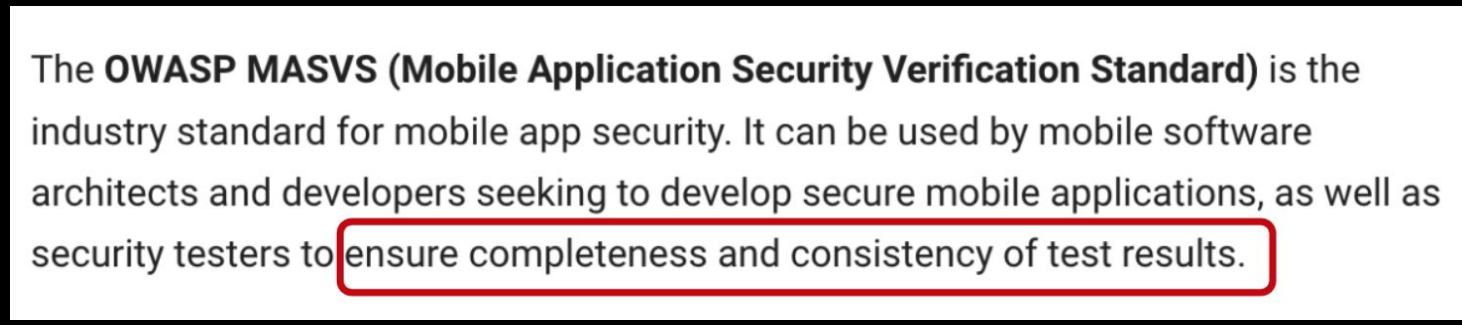
- Many students knew they would like to become pentesters.
- Mostly interested into web, mobile applications, and IoT.
- How to systematically test these?
- One of my students even contributed to the OWASP MASTG.

Pro: App Testing Guides

- Common issues that should be tested for every app.
- Beginner-friendly hints where to start testing.
- List of tools and how to use them for these tests (MASTG).

Contra: App Testing Guides

- Your results will never be complete.
- Consistency highly varies, as app developers use different programming frameworks. Applying the same tooling to the same bugs might still not reveal them in all cases.
- Bug classes and testing tools change over time.



Teach students to research, and they'll find bugs for a lifetime.

App Testing Guides Blind Spots

Threat Modeling & Security Testing

- Create your own list of threats.
- Prioritize what to test.
- There is no “out of scope”.
- Build new tooling when necessary.

OWASP MASVS:
Here are common application threats.

OWASP MASTG:
You can use the following tools for testing.

Mobile Apps = Web Clients?

OWASP MASVS treats many mobile application aspects similar to web clients.

Local vs. Cloud Storage

- Protect user data on mobile devices:
  - Avoid storing data on the phone.
  - Prevent data from being backuped onto another device.

- Where else should data be stored?

- Personal photos, fitness tracking, private messages…

  … local storage and local backup might actually be the best location!

MASVS-STORAGE

Reverse Engineering & Tampering

- Recommendation to prevent app reversing and manipulation.

  Existing tools might be difficult to bypass, but not impossible due to missing root of trust on jailbroken/rooted devices.

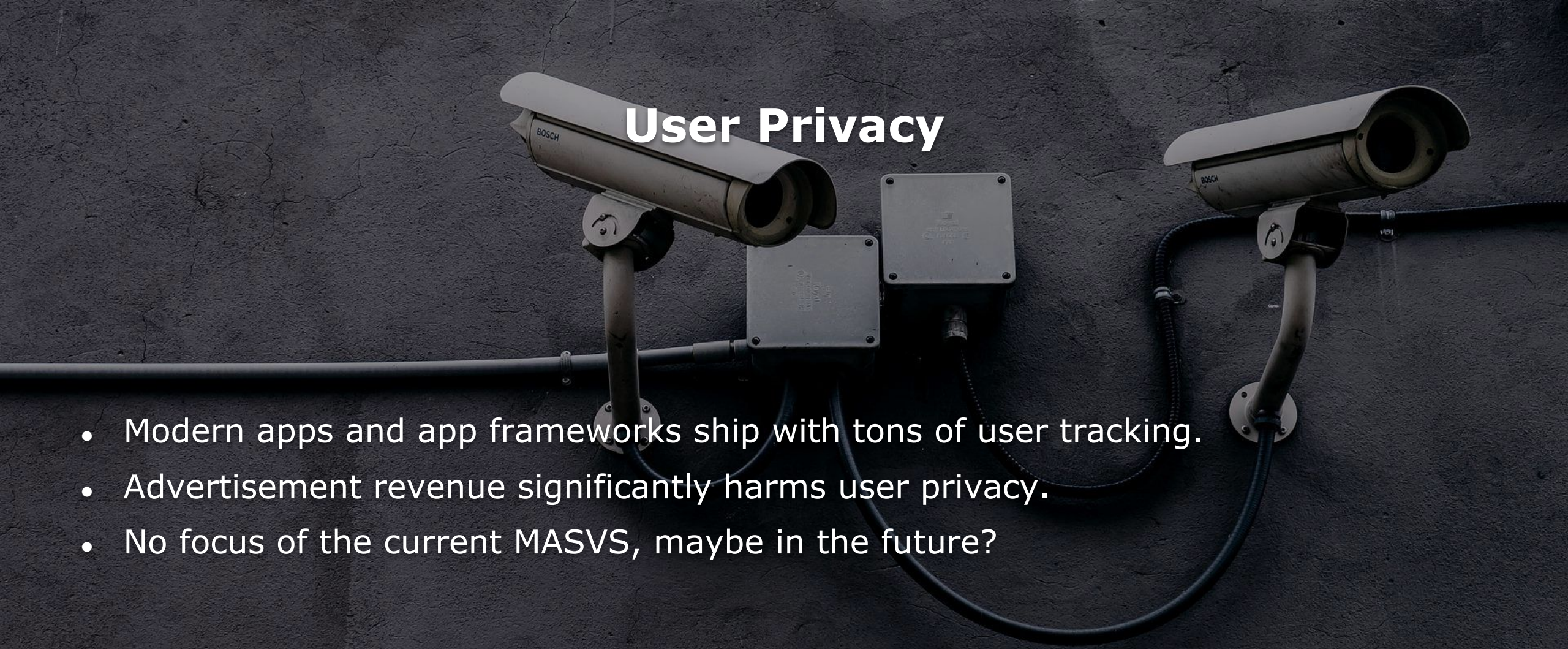
- Honest security researchers might no longer invest the time looking for bugs and reporting them via a bug bounty program.

- **This should never be the only security measure.**

  “Pentest done, we couldn’t bypass the jailbreak detection, app is secure.”

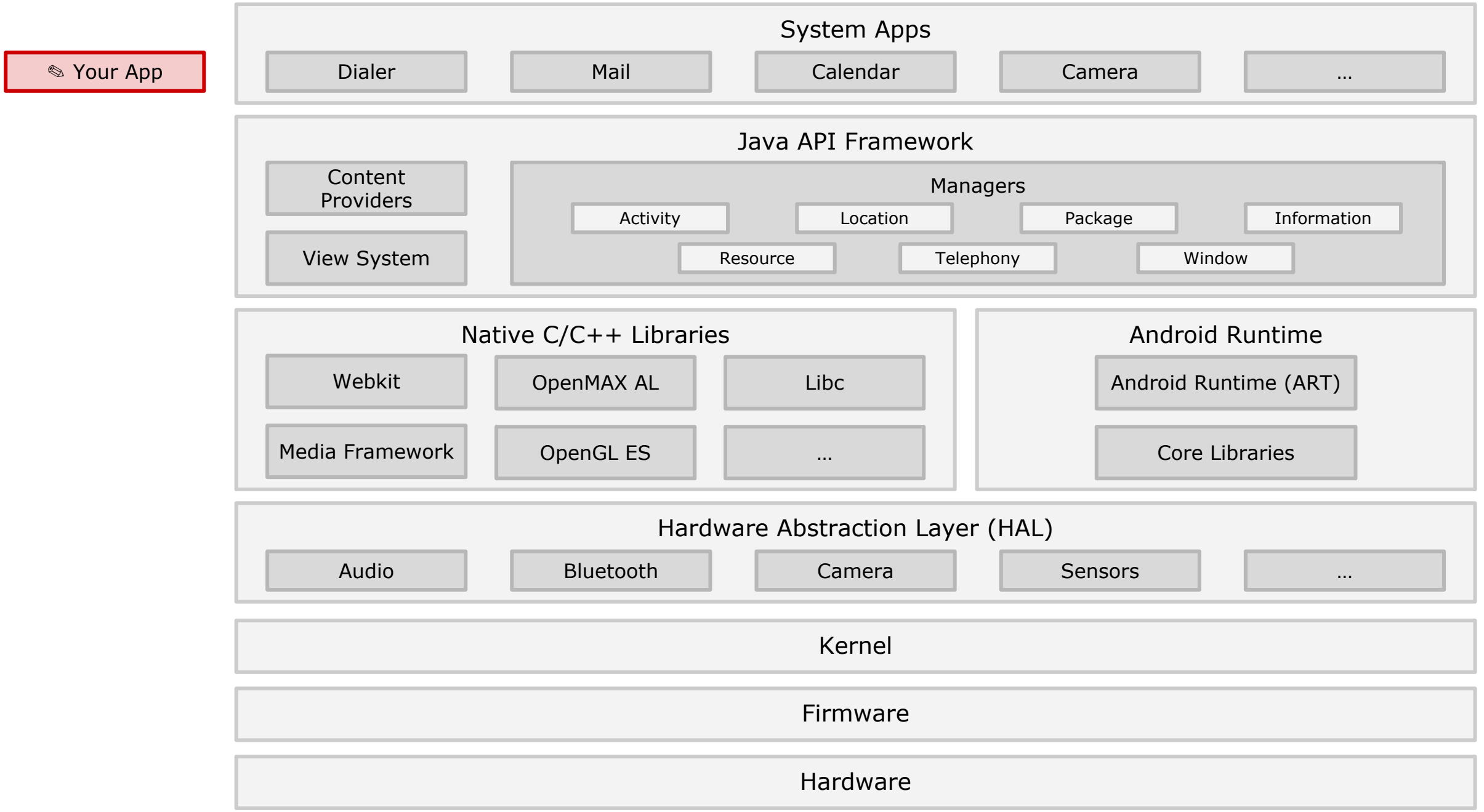
MASVS-RESILIENCE

User Privacy



- Modern apps and app frameworks ship with tons of user tracking.
- Advertisement revenue significantly harms user privacy.
- No focus of the current MASVS, maybe in the future?

MASVS-???


✎ Your App
System Apps
Dialer
Mail
Calendar
Camera
...
Java API Framework
Content Providers
View System
Managers
Activity
Location
Package
Information
Resource
Telephony
Window
Native C/C++ Libraries
Webkit
OpenMAX AL
Libc
Media Framework
OpenGL ES
...
Android Runtime
Android Runtime (ART)
Core Libraries
Hardware Abstraction Layer (HAL)
Audio
Bluetooth
Camera
Sensors
...
Kernel
Firmware
Hardware

Exploiting Frameworks

- Apps depend on iOS/Android & 3rd-party frameworks.
- Finding flaws in these could uncover issues affecting many apps or even the OS.
- Try testing “out of scope”, interesting bugs are waiting here!

Firmware Attacks

- Tradeoff between performance & security, especially on SoCs.
- Wireless chips: Where security measures from the 90ies meet modern smartphones.
- When designing an IoT ecosystem, also look into firmware security.

Hardware Attacks

- What if software is insecure due to the underlying hardware?
- **Side Channels**
  Obtain additional information that shouldn’t be accessible,
  e.g., cryptographic keys by monitoring execution time or power traces.
- **Fault Injection**
  Manipulate or skip instructions executed on the physical CPU,
  e.g., with voltage/clock glitching, electromagnetic waves, laser beams, …
- When designing new systems, keep the possibility of hardware attacks in mind.

MASVS and MASTG help identifying common issues.

Are these issues always significant security threats?

Will fixing them improve security?

Under which threat model?

Q&A

youtube.com/@jiskac


@jiska@chaos.social


github.com/seemoo-lab


jclassen@seemoo.de