Privacy Design Flaws

Eivind Arvesen, Group Cyber Security Manager (Sector Alarm)

PEPR ’22 – Conference on Privacy Engineering Practice and Respect
June 23rd 2022

EivindArvesen.com

@EivindArvesen

Background

Background

Flaws VS Bugs

«Flaws *in the design*» VS «Bugs in the code»

Background

Flaws VS Bugs

«Flaws *in the design*» VS «Bugs in the code»

Avoiding the Top 10 Software Security Design Flaws
https://ieeecs-media.computer.org/media/technical-activities/CYBSI/docs/Top-10-Flaws.pdf

Background

Motivation

- Privacy as an emergent property of the system in question
- Lack of explicit best practices that are concrete, actionable for technical roles that are not privacy experts
- Make privacy engineering basics common knowledge amongst developers & architects
- To enable easier discussion of architectural defects between technical privacy roles

Background

Two types of risks

Privacy risks are risks that you manage *on behalf of* data subjects – who are the ones who will be directly affected by the consequences if any risks are realized!

The flaws

The Flaws
From the CFP

- False anonymity
- Data leakage
- Mistaking data protection for privacy
- Failing to consider contextual requirements
- Unclear or changing purposes
- Assumed trust
- Misunderstanding data and definitions
- Insufficient data minimization
- Failure of protective controls
- Ethical issues

The flaws

Mistaking data protection for privacy

Description: Believing that a solution or system is privacy friendly by virtue of securing its data well.

Identify: Ask yourself whether you're able to explain why the solution is privacy friendly without referencing security controls.

Example: Smittestopp (v. 1)

- Aggressive data collection
- Breaking with regulatory requirements + best practice
- Argued that the parties involved were trustworthy – hence there was no issue

Avoid: Build competence, assess purpose limitation, lawful basis, degree of data minimization, …

The flaws

False anonymity

Description: Misinterpreting risks around (deidentified) personal data

Identify: Risk-assessment

Example: Researchers de-anonomizing/re-identifying users in the Netflix Prize Dataset:

> Researchers deanonymized the users in Netflix Prize dataset, which contained anonymous movie ratings of 500,000 subscribers – by correlating with IMDB data (knowing only a tiny bit about each person from before), uncovering their apparent political preferences and other potentially sensitive information.

Avoid: Err on the side of caution wrt. anonymization techniques.

The flaws

Assumed trust

Description: System implicitly builds on / assumes trust.

Identify: Threat modeling

Example: Facebook/Cambridge Analytica

> CA built psychological profiles of Facebook users to sell individual psychological targeting as a service, via a personality quiz. This app was able to obtain unusually rich info about users' friends via FB's Graph API because of permissive API scopes (TOS: only to help improve in app experience)

Avoid: Limitations need enforcing. Principle of least privilege.

The flaws

Data leakage

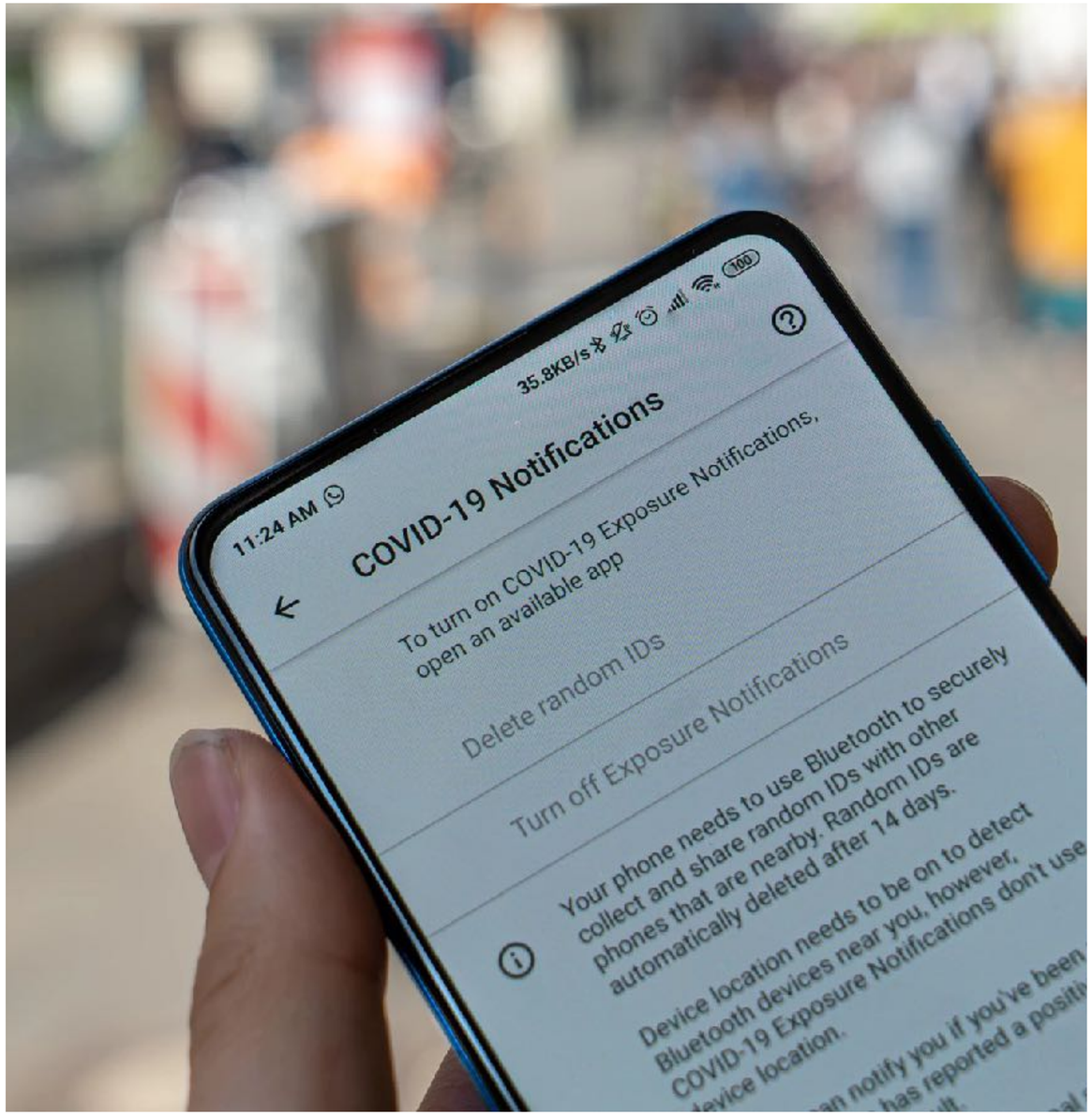
Description: System makes sensitive data available unintentionally

Identify: Threat modeling – what would attackers actually be able to do (vs. what you would expect your users to do)

Example: Android logging contact tracing apps information in system logs

> Privacy preserving, Rolling proximity identifiers (sent in BLE advertisements) – which are anonymous, but can be re-derived locally from Diagnosis keys (shared upon positive COVID diagnosis) based on RPI's you have met over a previous time window – were logged locally.

Avoid: Data classification schemes and policies; Logging policy (particularly wrt sensitive data).



Discussion & Conclusion

Discussion

The problem, summarized

We see that:

- There are multiple classes of generalizable privacy defects
- Some of these flaws result in bad outcomes, recognized from Security
- The basics of Privacy Engineering do not yet seem to be widely disseminated – is at least not foundational Software Engineering / Architecture knowledge

Discussion

Privacy by design (Ann Cavoukian, 1995)

1. Proactive not reactive; preventive not remedial
2. Privacy as the default setting
3. Privacy embedded into design
4. Full functionality – positive-sum, not zero-sum
5. End-to-end security – full lifecycle protection
6. Visibility and transparency – keep it open
7. Respect for user privacy – keep it user-centric

Conclusion

Solutions – Principles to abide by

- Privacy by design
- Architectural risk analysis; threat modeling
- Developing taxonomies, cheatsheets, standards, design patterns and architectural references

Conclusion

Source of Inspiration

- «Avoiding the Top 10 Software Security Design Flaws» – IEEE Center for Secure Design
- OWASP Top Ten
- OWASP Application Security Verification Standard

- ISO 27001
- NIST Cybersecurity Framework

Conclusion

Resources

- OWASP Top 10 Privacy Risks
- Privacy Patterns (UC Berkeley)
- LINDDUN (Threat modeling methodology)

- ISO 27701
- NIST Privacy Framework

Thanks!

Eivind Arvesen
 @EivindArvesen (Twitter)

eivind.arvesen@gmail.com
https://www.EivindArvesen.com