Privacy Engineering: Definitions and Challenges

IFIP Summer School 2017
Ispra (IT), Sep 6, 2017

José M. del Álamo, Yod-Samuel Martín
Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Contents

• Privacy by Design
• What’s Privacy Engineering?
• Privacy Engineering metamodel
• (Some) Privacy Engineering elements
• Conclusion and challenges ahead
Gap between research and practice

Privacy (research)
- Theories
- Paradigms
- Frameworks
- Threats and risks
- Technologies

(Software) engineering
- Development process
- Business domain
- System type

Privacy by Design
Privacy by Design

1. **Proactive** not Reactive; **Preventative** 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**

*Source: Ann Cavoukian, Privacy By Design – The 7 Foundational Principles (2011)*
Software Development Lifecycle

Product Backlog → Sprint Backlog → Sprint → Working increment of the software

- 24 h
- 30 days

Engineering privacy?

- **Software/Systems engineers:**
  - Primarily focus on meeting functional requirements
  - Consider some quality attributes depending on the domain: performance, security, safety...
  - (Usuallu) Are not know aware of legal concepts
  - Are not familiar with all the (complex) privacy technologies

- **Instead:**
  - "feel that privacy is an abstract problem, not an immediate problem, not their problem, or not a problem at all"

The non-compliance costs (reactive + corrective measures) is 3 times greater than the compliance costs.

The costs of change in a software development process increases as the process advances.

Contents

• Privacy by Design
• What’s Privacy Engineering?
• Privacy Engineering metamodel
• Privacy Engineering elements
• Conclusion and challenges ahead
Privacy engineering is an emerging research framework that focuses on designing, implementing, adapting, and evaluating theories, methods, techniques, and tools to systematically capture and address privacy issues in the development of sociotechnical systems.

Seda Gürses & Jose M. del Alamo
Privacy Engineering: Shaping an Emerging Field of Research and Practice
Privacy Engineering

Field of research and practice that designs, implements, adapts, and evaluates theories, methods, techniques, and tools to systematically capture and address privacy issues when developing information systems.

Theory: Privacy conceptualization
Privacy Engineering

Field of research and practice that designs, implements, adapts, and evaluates theories, methods, techniques, and tools to systematically capture and address privacy issues when developing information systems.

Method: Process description providing directions and rules and helping to set privacy goals, structured in a systematic way in tasks and stages.
Privacy Engineering

Field of research and practice that designs, implements, adapts, and evaluates theories, methods, techniques, and tools to systematically capture and address privacy issues when developing information systems.

Technique: Procedures, possibly with a prescribed language or notation, to accomplish specific privacy-engineering tasks.

T. Breaux et al., Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements.
Privacy Engineering

Field of research and practice that designs, implements, adapts, and evaluates theories, methods, techniques, and tools to systematically capture and address privacy issues when developing information systems.

Tools: Means (automated or not) that support privacy engineers to carry out their responsibilities within a privacy-engineering method.
Contents

• Privacy by Design
• What’s Privacy Engineering?
• Privacy Engineering metamodel
• Privacy Engineering elements
• Conclusion and challenges ahead
“Privacy Engineering”

Research publications (Scopus)
Privacy Engineering landscape?

Picture: Christian Lopez (the garage), https://www.flickr.com/photos/andidigress/4471100232
A Privacy Engineering Methodology Metamodel

Picture: Justin de la Ornellas (avex2) https://www.flickr.com/photos/ornellas/2835160463/
Method engineering: How are methodologies defined?

**Metamodel**
- Any methodology is made of different types of elements.
- The description of any methodology is based on a set of concepts.
- Instances of these concepts may be related with one another.
- E.g. Any methodology is made of Tasks, Processes, Roles, Documents...

**Methodology**
- Instantiates a metamodel by defining specific instances of elements and relations.
- Different methodologies conform to the same, shared metamodel.
- E.g. A specific methodology defines a Requirements Analysis Task which yields a Requirements Document, etc.

**Endeavour**
- Enacts the methodology during each project.
Privacy Engineering revisited

Privacy Engineering contributions define **Method(ologie)s** or Methodological elements (**Fragments**) to systematically capture and address privacy issues in the development of information systems made of:

**Producers** (role, tool or team) who perform some **Work Units** (process [goal], task [what], technique [how]) which **act upon** (reads, creates, modifies, deletes) some **Work Products** (documents, models*, software items) while at a **Stage** (at a milestone, or during a phase, build, or time cycle) and using some **given Resources** (languages, notations, guidelines and constraints).

*Models are composed of Model Units, conform to a Language, and are expressed in a Notation*
Privacy Engineering Metamodel
Privacy Engineering Resources

Ontological perspective

Privacy Conceptual Units
(e.g. principles, harms, goals..)

Model Units
(e.g. patterns, strategies, threats) w.r.t. Language

Deontological perspective

- Existential Constraints
- Temporal Constraints
- Privacy Endeavor Requirements

Guidelines (in a context)

Situational perspective

Privacy Conceptual Model

Privacy Normative Framework

Privacy Knowledge Base

Privacy Engineering Code

Epistemological perspective

Existential Constraints

Temporal Constraints

Privacy Endeavor Requirements

Guidelines (in a context)

Situational perspective
LINDDUN as a Privacy Engineering Method

PROCESS: ELICITATION OF PRIVACY THREATS

1. Task: Define data flow
   - Tech.: Create DFD from requirements
     reads System Requirements Specification
     creates Data Flow Diagram
   - Tech.: Create DFD from Architecture
     reads System Architecture Document
     creates Data Flow Diagram

2. Task: Map threats to data flow elements
   - Tech.: Use threat mapping template
     reads Data Flow Diagram
     creates Threat Map. table
   - Tech.: Discard less likely threats
     modifies Threat Mapping Table
   - Tech.: Combine threats (‘reduction’)
     modifies Threat Mapping Table

3. Task: Elicit privacy threats
   - Tech.: Refine threats
     reads Threat Mapping Table
     creates Threat List
     uses Privacy Threat Tree Catalogue
   - Tech.: Document assumptions
     creates Assumption List
     modifies Threat List
   - Tech.: Document threats
     reads Threat List
     creates Misuse Cases

PROCESS: SELECTION OF MITIGATING SOLUTIONS

4. Task: Prioritize threats
   reads Risk Assessment Document
   modifies Threat List

5. Task: Elicit mitigation strategies
   - Technique: Map threats to strategies
     reads Threat List
     creates Mitigation Strategies List
     uses Mitigation Strategies (Taxonomy & Mapping)

6. Task: Select Privacy-Enhancing Techniques
   - Technique: Map strategies to solutions
     reads Mitigation Strategies List
     creates PETs List
     uses Privacy-Enhancing Solutions Catalogue
Contents

• Privacy by Design
• What’s Privacy Engineering?
• Privacy Engineering metamodel
• Privacy Engineering elements
• Conclusion and challenges ahead
PRIPARE methodology
Requirements and Risks
Privacy Design Strategies

Privacy Pattern

A privacy pattern is “a reliable, implementable way to meet needs (stakeholders’ or legal) embodied in privacy requirements”

They have been heuristically proved (through experience) that are a good solutions
Privacy Dashboard

Requirement: Users should be aware of all the data being collected, created, maintained, processed and shared by the service provider or third parties.

Control: A privacy dashboard provides a summary of the different types of personal data held by the service provider, together with user controls to restrict purposes or sharing, when applicable.
Privacy Patterns Catalogue

https://privacypatterns.org/

Website to collect and discuss about privacy patterns

privacypatterns.org

Privacy patterns are design solutions to common privacy problems — a way to translate "privacy-by-design" into practical advice for software engineering. We believe design patterns can help document common practices and standardize terminology and while we're starting with a set of patterns for location-based services, we hope to build a living community space where all can contribute their own patterns.

Learn More »

Popular Patterns

Privacy-Aware Network Client
A privacy policy which is hard to understand is in an automated way converted into a more easy to read format.

Location Granularity
Support minimization of data collection and distribution. Important when a service is collecting location data from or about a user, or transmitting location data about a user to a third-party.

use of dummies
This pattern hides the actions taken by a user by adding fake actions that are indistinguishable from real.
PETs Assessment Repository

Source: M. Hansen, J.-H. Hoepman, M. Jensen (2016), Readiness Analysis for the Adoption and Evolution of PETs. ENISA.
Contents

• Privacy by Design
• What’s Privacy Engineering?
• Privacy Engineering metamodel
• Privacy Engineering elements
• Conclusion and challenges ahead
Summary

• **Privacy engineering** is the field of research and practice that seeks to provide systematic methods, techniques and tools to build privacy-friendly ICT systems

• Common conceptual framework
  ➔ **shared understanding** of concepts and relationships

• Methodologies defined in compatible terms
  ➔ easier integration of elements or fragments from different privacy and mainstream engineering methods
Challenges ahead

• Missalignment of SSH and technical backgrounds
  – How do we translate legal obligations into technical requirements?
  – How do we consider the social aspects?
• Rough transition between development phases
  – GDPR focuses on risks, but engineers tend to think on requirements
  – How do we move to design and then to implementation?
• Lack of supporting engineering tools/knowledge bases
• More (re)usable PETs
• New development practices: Agile, DevOps, XaaS...
• Training and education
• Standardization
  – Specially for catalogues and cheat sheets
The purpose of IPEN is to bring together developers and data protection experts with a technical background from different areas in order to launch and support projects that build privacy into everyday tools and develop new tools which can effectively protect and enhance our privacy.

IPEN objectives

The IPEN initiative was founded in 2014. It supports the creation of engineer groups working on stand other tools for selected internet use cases where...
International Workshop on Privacy Engineering

May 25

Welcome, introductions and opening remarks  9:00AM - 9:15AM

The Difffer Framework: Revisiting Noise, Again.
Invited talk by Paul Francis (Max Planck Institute for Software Systems)  9:15AM - 10:15AM

Break (30 Minutes)  10:15AM - 10:45AM

Session 1: Privacy engineering evaluation  10:45AM - 11:35PM

Click on any session to see more information. The program is also available as a PDF here.

http://www.ieee-security.org/TC/SPW2017/IWPE
Thank you

Jose M. Del Alamo
jm.delalamo@upm.es