# Knowledge is Power: Systematic Reuse of Privacy Knowledge for Threat Elicitation

Kim Wuyts, Laurens Sion, Dimitri Van Landuyt, Wouter Joosen

*imec-DistriNet, KU Leuven*
Heverlee, Belgium
{kim.wuyts, laurens.sion, dimitri.vanlanduyt, wouter.joosen}@cs.kuleuven.be

***Abstract*—Privacy threat modeling is difficult. Identifying relevant threats that cause privacy harm requires an extensive assessment of common potential privacy issues for all elements in the system-under-analysis. In practice, the outcome of a threat modeling exercise thus strongly depends on the level of experience and expertise of the analyst. However, capturing (at least part of) this privacy expertise in a reusable threat knowledge base (i.e. an inventory of common threat types), such as LINDDUN's and STRIDE's threat trees, can greatly improve the efficiency of the threat elicitation process and the overall quality of identified threats. In this paper, we highlight the problems of current knowledge bases, such as limited semantics and lack of instantiation logic, and discuss the requirements for a privacy threat knowledge base that streamlines threat elicitation efforts.**

***Index Terms*—privacy, threat modeling, knowledge base**

## I. Introduction

Privacy threat modeling is the systematic analysis of potential privacy threats in a software architecture. Especially with the GDPR [1] enforcing data protection impact assessments, such threat modeling exercises are indispensable for fulfilling the technical part of the obligated impact assessment by eliciting, assessing, and mitigating architectural privacy threats. Threat modeling [2]–[4] is, however, a quite labor-intensive and time-consuming activity. Both a technique and a repertoire (i.e. a set of threat types) are essential [3]. It does not only require a solid understanding of the system-under-analysis, expert privacy knowledge is essential in order to systematically analyze the system for potential privacy harm.

The elicitation step requires as input a set of *threat types* (i.e. potential privacy issues). This knowledge can be provided by a privacy expert, but it would clearly be more cost-efficient to capture (at least part of) this privacy expertise and provide it as reusable knowledge during threat elicitation. *Threat knowledge bases* (i.e. collections of threat types), such as LINDDUN [5], [6], STRIDE  [2], [7], CAPEC [8], CWE [9], OWASP's top 10 [10] and CNIL's threat list [11], can also be consulted. They however do not sufficiently support all threat modeling needs (e.g. they only have limited semantical support, lack instantiation logic to scope the knowledge to the analyst's needs, etc). In this paper, we will identify shortcomings of current knowledge bases and lay out the requirements for privacy threat knowledge bases to enhance the efficiency, quality and reproducibility of the threat elicitation process.

In Section II, we provide some background information on privacy threat modeling. We propose a set of requirements to streamline threat elicitation efforts in Section III. Section IV discusses the limitations of available privacy and security threat knowledge bases. Section V wraps up the paper with a discussion and conclusion.

## II. Threat Modeling

Threat modeling [3] is the systematic elicitation and mitigation of threats in software architectures. It originally gained traction in the security domain [2], but also privacy-focused threat modeling frameworks have emerged [4], [12].

Threat modeling generally consists of two phases. The *problem-oriented* phase comprises modeling the system and systematically eliciting threats. In the *solution-oriented* phase, the identified threats are assessed and mitigated. In this paper, we focus primarily on the threat elicitation step.

*1) Model the system:* First, a model is created to define the system-under-analysis. Typically, a Data Flow Diagram (DFD) [13] notation is used, which makes use of 5 element types: *external entities* (i.e. users or third party services external to the system), *data stores* (i.e. passive containers of information), *processes* (i.e. computational units), *data flows* (i.e. communication between DFD elements), and *trust boundaries* (i.e. a logical or physical division of the system).

*2) Elicit Threats:* The system is systematically analyzed by iteratively examining each system component. There are two main approaches. The original *per-element* approach [2], [4] iterates systematically over each element of the model, while the *per-interaction* approach [3], [14] considers each interaction (i.e. sender-flow-destination combination) in the model. For each element or interaction, applicable threats need to be identified. To determine which threats are applicable, potential privacy issues need to be known. One can rely on the knowledge of the team's privacy expert, but this privacy expertise can also be captured in a privacy knowledge base.

A *privacy threat knowledge base* consists of common *privacy threat types* (i.e. potential privacy issues). Existing knowledge bases are discussed in Section IV. Evidently, not all threat types apply to each element or interaction of the model. Database-specific threat types should for instance not be examined for a

This research is partially funded by the Research Fund KU Leuven and the PRiSE KU Leuven-C2 research project.

© 2019, Kim Wuyts. Under license to IEEE.
DOI 10.1109/SPW.2019.00025

flow or external entity, and vice versa. The presented knowledge should thus be scoped to those threat types that are relevant for the specific elicitation iteration.

*3) Assess, Prioritize, and Mitigate Threats:* The remainder of the threat modeling methodology consists of tackling the identified threats. First, the threats are assessed and prioritized according to their risk. Second, each threat is systematically addressed by selecting suitable privacy solutions.

## III. Requirements for Knowledge Support

Based on ongoing efforts to streamline the threat modeling practice, we discuss the requirements for a privacy threat knowledge base that facilitates threat elicitation efforts.

### *A. Semantics of Knowledge Base (Structure)*

Each *threat type* should provided the same type of information, and contain at least:

*Description:* contains, at least, the *explanation* of the threat type, a title, an *identifier* (for quick reference), and some *examples* to make the threat type more tangible.

*Precondition(s):* describes the conditions that can lead to the threat type. Also, it should be clear whether all conditions are required or not (`and`- vs. `or`-relationship).

*Relationship(s):* can exist with other threat types. This should include *impact* relationships (e.g., a consequence) and *hierarchical* relationships between threat types of different abstraction levels (which allows the documentation of both high-level and more detailed and applied threat types).

These semantics will force the knowledge base creator to thoroughly reason about each threat type and hence fill potential gaps in knowledge of, and identify relationships among, different threat types.

### *B. Support for Instantiation Logic*

For efficient threat elicitation, instantiating a knowledge base perspective targeted at specific system, threat and application properties allows to iteratively focus on those threats types that are applicable to the component being analyzed.

*System properties:* The threat modeling process typically iterates over each system component (or interaction) individually. Instantiating the knowledge base with threat types specifically targeted for that component (e.g., selection based on DFD element type, interaction type, data type, etc.) would greatly improve the relevance of the provided knowledge.

*Threat properties:* The analyst can prefer to iterate over threat categories rather than system components. Also, the analyst might determine upfront that certain threat types or categories are (not) important for the system-under-analysis. For instance, by specifying the need for (or exclusion of) threats to user anonymity (i.e. anonymous authentication and communication), the knowledge base can be better scoped.

*Application properties:* Application- and domain-specific threat type refinements can also be included in the knowledge base (e.g., threat types specific for IoT applications). In addition, the application context can predetermine certain threat properties (or lack thereof). For example, certain applications (such as e-voting systems) will require user anonymity, while 'corporate' applications typically do not.

Instantiating a knowledge base view based on these properties largely increases the relevance and applicability of threat types to be examined and hence the efficiency of the process [6]. To fully support this instantiation logic, the knowledge base structure (Section III-A) should reflect these selection criteria (i.e. as precondition sub-types).

### *C. Integration with Solution Phase*

The knowledge base can also assist in later threat modeling steps. Each threat type can already include information on the degree it will influence the likelihood and impact in the risk assessment step or a threat type specific method to assess risk.

Similarly, the dependency between threat types and solutions can be integrated. Threat types can list solutions to mitigate them. Also, the impact of a solution on each threat type is valuable, as a system might already have a number of security and privacy solutions in place. Being able to focus specifically on those threat types that still apply despite the implementation of a certain solution (e.g., encrypted communication, user authentication, etc.) can greatly reduce the number of irrelevant threat types that need to be evaluated in the knowledge base [6].

These concepts should also be reflected in the semantics of the knowledge base (Section III-A).

### *D. Extensibility*

Requirements for threat elicitation can evolve. For instance, new architectural styles can require a different approach. Also, when the privacy domain further progresses (and new solutions or threat types emerge), there might be a need to support more complex threats, for instance, a threat type specific to a chain of interactions (e.g., a threat type that applies to the entire single-sign-on interaction chain). These advances should also be supported by the knowledge base. In addition, each analyst will have its own domain-, application-, and company-specific expertise that should be systematically reusable. Both the knowledge base structure and the knowledge base itself should be easily extendable, as technology is ever evolving.

### *E. Support for Alternating Audience and Abstraction Levels*

The principal goal of threat modeling is a (close to) complete analysis of threats by systematically evaluating each component of the system in detail. However, in practice, given varying time constraints and levels of expertise, different threat modeling approaches are executed. The essence of the process –identifying (and mitigating) relevant threats– remains but the thoroughness of each iteration over system components and the reusable knowledge is in practice tailored to the project at hand. Sometimes only the abstract threat types (e.g., the high-level threat categories encompassed in the STRIDE and LINDDUN acronyms) are used as input for a more informal brainstorm. Or, only the concrete examples are used, as some prefer more tangible inputs over abstract descriptions. Alternatively, a privacy expert will still systematically analyze each system component, but might not require to assess all threat types in

TABLE I
EVALUATION OF EXISTING KNOWLEDGE BASES W.R.T. PROPOSED REQUIREMENTS

| | **Semantics** | | | **Selection criteria** | | | **Solution integration** | **Extensibility** | **Abstraction level** |
|---|---|---|---|---|---|---|---|---|---|
| | Description | Conditions | Relations | System properties | Threat properties | Application properties | | | |
| **LINDDUN** [4], [12] | ● | ◒ | ◒ | ◒ | ◒ | ○ | ○ | ○ | ○ |
| **LINDDUN**$^{ref}$ [6] † | ● | ◒ | ◒ | ◒ | ◒ | ◒ | ○ | ○ | ○ |
| **STRIDE** [2], [7] * | ● | ◒ | ◒ | ◒ | ◒ | ◒ | ○ | ◒ | ○ |
| **CWE** [9] | ● | ◒ | ◒ | ○ | ◒ | ○ | ◒ | ○ | ○ |
| **CAPEC** [8] | ● | ◒ | ◒ | ○ | ◒ | ○ | ◒ | ○ | ○ |
| **OWASP** [10] | ● | ● | ○ | ○ | ○ | ○ | ◒ | ○ | ○ |
| **CNIL** [11] | ◒ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

*Legend:* ● *: full support,* ◒ *: partially supported,* ○ *: no support.*
† *LINDDUN*$^{ref}$ *is an extension of LINDDUN that uses domain-specific refinements to streamline threat elicitation [6].*
* *STRIDE is evaluated as the combination of its threat tree catalog [2] and its tool [7].*

the knowledge base. A threat knowledge base should provide knowledge at the suitable abstraction level for each of these scenarios, by, for instance, allowing an extraction of the required content per use case.

Similarly, depending on the audience and use case, different support is required. Full-fletched analysis would greatly benefit from (semi-)automated tool support, while for more lightweight threat modeling approaches an (interactive) catalog is likely preferred. It is thus required that different output formats can be generated depending on the use case (and audience), varying from a plain text, printable document to a more advanced interactive-style catalog for manual analysis and even a machine-readable format when integrated in threat modeling tool support.

A wide range of instantiation criteria (Section III-B) that are also reflected in the knowledge base structure (Section III-A) will be required to tune the knowledge base for these alternating audiences and abstraction levels.

## IV. THREAT KNOWLEDGE BASES

Several knowledge bases exist, but they each come with their own limitation with respect to threat modeling support. In this section, we inventorize and evaluate existing privacy and security threat knowledge bases with respect to the requirements discussed in Section III. For the evaluation (summarized in Table I), we apply the following scale: ● *: fully supported,* ◒ *:partially supported,* ○ *: not supported.*

### *A. LINDDUN & STRIDE - Privacy & Security Threat Catalogs*

*Description:* LINDDUN [4], [5] provides, inspired by STRIDE [2], its own reusable knowledge base, specifically targeted for systematic threat elicitation. It is presented as a *catalog of threat trees*, structured according to LINDDUN or STRIDE threat category and DFD element type. An example of such a threat tree is shown in Figure 1. Each tree represents the most common attack paths for that particular combination of LINDDUN or STRIDE threat category and DFD element type. The STRIDE threat types are also included (in XML format) in the SDL tool [7]. We evaluate STRIDE as the combination of its tool and catalog knowledge base.

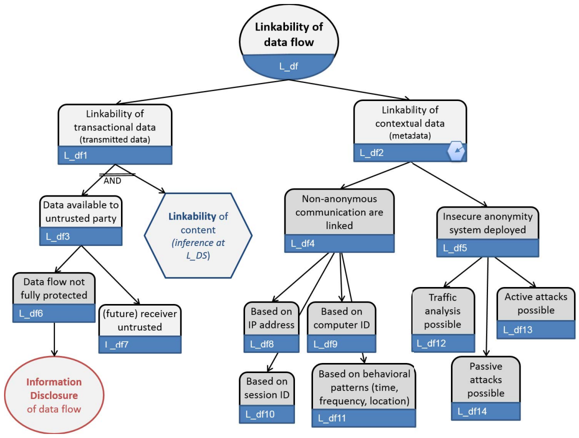
Linkability of data flow
L_df
Linkability of transactional data (transmitted data)
L_df1
Linkability of contextual data (metadata)
L_df2
AND
Data available to untrusted party
L_df3
Linkability of content (inference at L_DS)
Non-anonymous communication are linked
L_df4
Insecure anonymity system deployed
L_df5
Data flow not fully protected
L_df6
(future) receiver untrusted
L_df7
Information Disclosure of data flow
Based on IP address
L_df8
Based on computer ID
L_df9
Traffic analysis possible
L_df12
Active attacks possible
L_df13
Based on session ID
L_df10
Based on behavioral patterns (time, frequency, location)
L_df11
Passive attacks possible
L_df14

Fig. 1. Example of a LINDDUN threat tree: linkability of data flow (from Wuyts et al. [5])

*Evaluation:* As reflected in Table I, each threat has a *description*, and there is partial support for selection based on *threat properties* (i.e. the LINDDUN and STRIDE threat categories respectively) and on *system properties* (i.e. per DFD element type [2], [12] or per DFD interaction [7]). There is however semantical ambiguity in the tree structure: some nodes are actual threats (e.g., insecure anonymity system deployed ($L\_df5$) in Figure 1), while other are more like *preconditions* (e.g., untrusted (future) receiver ($L\_df7$)). Its tree format does enforce (hierarchical) *relationships* between threat types and threat categories, yet without a semantical foundation. There is currently no support for *solution integration* or multiple *abstraction levels*. Developments towards *application properties* selection are ongoing: an extension of LINDDUN [6] proposes the use of domain refinements to already partially tackle the issue (as shown by the second row in Table I) and the STRIDE tool [7] allows instantiation of application-specific DFD elements that can contribute to a more focused threat elicitation. With respect to *extensibility*, the XML representation of the STRIDE tool knowledge base can be easily updated and extended by the analyst.

### B. CAPEC & CWE - Security Attack & Vulnerability Catalogs

*Description:* CAPEC [8] is a *catalog* of common attack patterns, structured according to attack mechanism and attack domain. CWE [9] is a *catalog* of common weaknesses, structured according to research, development and architectural concepts. They provide security insights to software developers in how their systems are likely to be attacked.

*Evaluation:* They both have similar structure including a *description* of the problem, (partial) support for *preconditions* and *relationships*. They link to proposed *solutions* and can be structured according to generic *threat properties* and tactics. Their *system properties* (i.e. the architectural components where the problem would arise) are however not (clearly) described, making assessment of threat type applicable difficult. Also, there is no support for *application-specific selection*, *extensibility* or *multi-use case* scenarios.

### C. CNIL & OWASP - Threat Lists

*Description:* OWASP [10] publishes its top 10 op most critical web application security risks. CNIL [11] provides a list of generic confidentiality, integrity and availability threats.

*Evaluation:* CNIL and OWASP provide flat lists, meant to serve as general best practices rather than to systematically iterate over them. OWASP's list does provide a clear *structure* to describe each threat and proposes *solutions*. Further support of knowledge base requirements is lacking for both.

### D. Alternative Knowledge Representations

Knowledge inventories documenting 'positive' privacy requirements and goals also exist. Among others, the PriS method [15] uses privacy-process patterns to analyze the effect of privacy requirements on organizational processes. Beckers and Heisel [16] proposed a set of privacy requirements patterns. Oetzel and Spiekermann [17] provide in their methodology for privacy impact assessments a list of privacy targets. Regardless of the negative or positive approach, our proposed requirements also apply. They are (flat) lists of high-level requirements and therefore lack most of the requirements, similar to the threats lists in Section IV-C. As they are, by definition, no threat knowledge bases, they are however not included in the evaluation (in Table I).

## V. Discussion and Conclusion

Providing the right knowledge at the right time is key for efficient threat modeling. This paper proposes requirements for a threat knowledge base that will strengthen the threat elicitation process: knowledge base semantics, support for instantiation logic, solution integration, extensibility, and support for alternating audience and abstraction levels.

Note that these requirements alone do not guarantee a high-quality knowledge base. Aspects such as coverage, relevance, and applicability are also essential. In order to provide a precise description of each threat type, the underlying privacy concepts need to be clear. Privacy engineering is still a relatively young research domain. Although quite some work on privacy concepts such as anonymity [18]–[20] exist, they all tend to approach the concept differently. Despite their very extensive descriptions, there is still terminological ambiguity (e.g., varying definitions on 'identifiable data').

The requirements specified in this paper can largely reduce the threat elicitation effort (as well as the knowledge base maintenance). Our evaluation has however shown that current knowledge bases do not sufficiently support this pentad of requirements. Therefore, the ongoing improvements of LINDDUN will include a revision of the knowledge base in accordance with these requirements by defining a sound meta-model for the knowledge base and instantiating it with threat types founded on a solid conceptual privacy model.

Overall, a privacy threat knowledge base that harmonizes the proposed requirements will support systematic reuse of privacy threat knowledge and will thus lower the dependence on intangible expertise. This will improve efficiency and reproducibility of the threat modeling process, and pave the way towards tool support for (semi-)automated threat elicitation.

## References

[1] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 ," *Official Journal of the European Union*, 2016.

[2] M. Howard and S. Lipner, *The Security Development Lifecycle*, 2006.

[3] A. Shostack, *Threat Modeling: Designing for Security*, 2014.

[4] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, 2011.

[5] K. Wuyts, R. Scandariato, and W. Joosen, "LIND(D)UN privacy threat tree catalog (CW675)," Dept. of Computer Science, KU Leuven, 2014.

[6] K. Wuyts, D. Van Landuyt, A. Hovsepyan, and W. Joosen, "Effective and Efficient Privacy Threat Modeling through Domain Refinements," in *Proceedings of the ACM Symposium on Applied Computing*, 2018.

[7] M. Corporation, "Microsoft threat modeling tool 2016," 2016. [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=49168

[8] "Common Attack Pattern Enumeration and Classification (CAPEC)." MITRE. [Online]. Available: https://capec.mitre.org/

[9] "Common Weakness Enumeration (CWE)." MITRE. [Online]. Available: https://cwe.mitre.org/

[10] OWASP, "OWASP top 10 - 2017: The ten most critical web application security risks."

[11] CNIL, "Methodology for privacy risk management: How to implement the Data Protection Act."

[12] K. Wuyts, "Privacy Threats in Software Architectures," Ph.D. dissertation, KU Leuven, jan 2015.

[13] T. DeMarco, *Structured Analysis and System Specification*, 1979.

[14] L. Sion, K. Wuyts, K. Yskout, D. Van Landuyt, and W. Joosen, "Interaction-based privacy threat elicitation," in *International Workshop on Privacy Engineering*, 2018.

[15] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," *Requirements Engineering*, 2008.

[16] K. Beckers and M. Heisel, "A foundation for requirements analysis of privacy preserving software," *Lecture Notes in Computer Science*, vol. 7465 LNCS, pp. 93–107, 2012.

[17] M. C. Oetzel and S. Spiekermann, "A systematic methodology for privacy impact assessments: A design science approach," *European Journal of Information Systems*, vol. 23, no. 2, pp. 126–150, 2014.

[18] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010, v0.34.

[19] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[20] S. L. Garfinkel, "De-identification of personal information," *NISTIR*, vol. 8053.