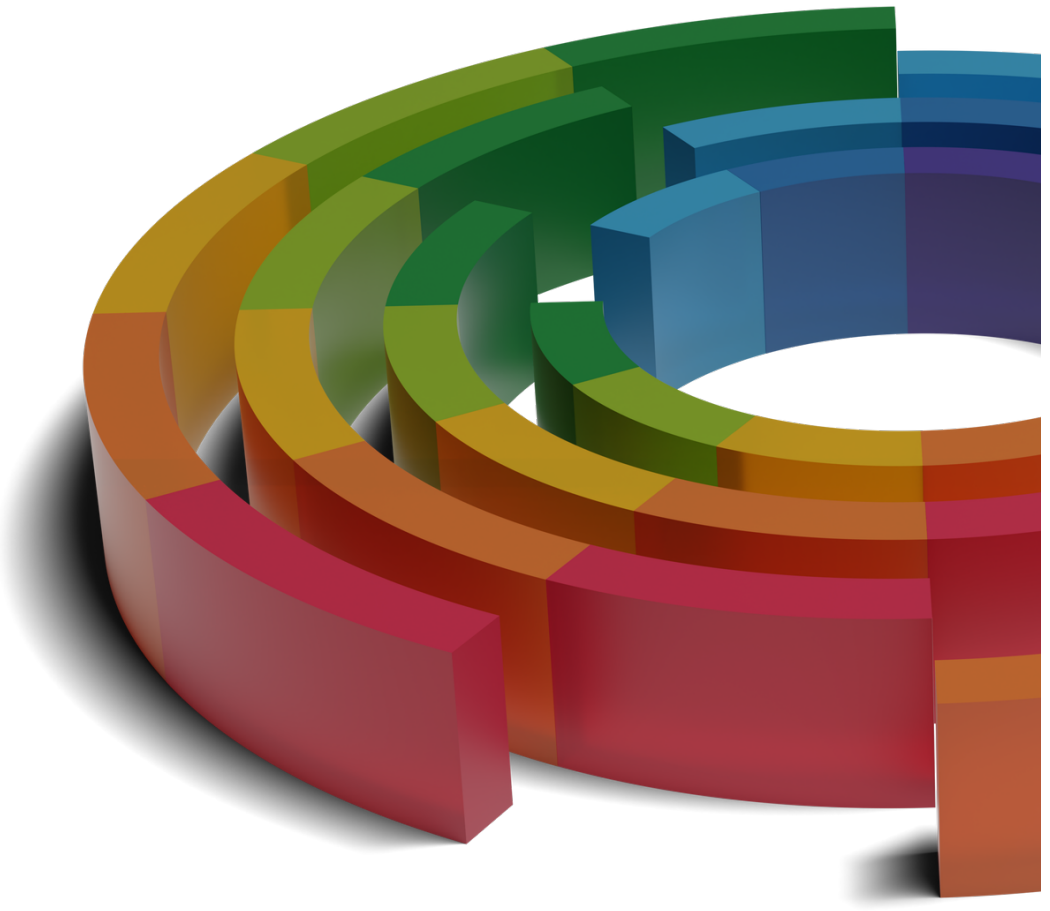
LINDDUN PRO

PRIVACY THREAT MODELING TUTORIAL



DistriNet
DISTRINET, KU LEUVEN

LINDDUN

This is a pre-release version of the LINDDUN PRO Privacy Threat Modeling Tutorial.
The latest version can be found at:
`https://downloads.linddun.org/tutorials/pro/latest/tutorial.pdf`

WWW.LINDDUN.ORG/PRO

Version 0.1, April 2023

Please refer to this document as:
Laurens Sion and Wouter Joosen, **LINDDUN PRO privacy threat modeling tutorial**, Technical Report, Department of Computer Science, KU Leuven, April 2023

Contents

1 *Overview* 7

2 *LINDDUN Privacy Threat Types* 9

3 *Creating the DFD model* 13

4 *Eliciting threats* 17

5 *Documenting and prioritizing threats* 24

Bibliography 29

Introduction

Privacy is an important quality of software systems and it is increasingly recognized as a property that has to be considered from the start when designing and building software systems. This is emphasized with the term Privacy by Design.[1] This is not only desirable property from the end-user or organizational perspective, there is a lot of legislation around the world that forces organizations that process personal data to consider privacy and data protection concerns in the design and development of their software systems.[2] Therefore, the use of techniques such as privacy threat modeling can also help you meet some of these compliance requirements.

LINDDUN PRO is a threat modeling approach that can help you to address privacy concerns in the software that you are building by doing a systematic analysis of your software design to identify potential privacy threats. By systematically eliciting and addressing the uncovered privacy threats from the early phases of development, you can build more privacy-friendly software systems from the start, rather than reactively fixing privacy problems later on. This, of course, does not prevent you from doing these analyses for your existing systems.[3] But the problems you identify may be tougher, more complex, or expensive to fix in existing systems.

[1] Ann Cavoukian. Privacy by design [leading edge]. *IEEE Technology and Society Magazine*, 31(4):18–19, 2012. DOI: 10.1109/MTS.2012.2225459

[2] European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union*, 59(L 119):1–88, May 2016

[3] On the contrary, it can be very insightful to perform these analyses on existing systems as well.

Who is this document for?

This document targets software engineers that want to analyze the privacy of the software systems that they are building. The tutorial does not require any detailed background knowledge on privacy or modeling. Chapter 2 will provide the necessary background information on the LINDDUN privacy threat types that you will use during the analysis. While it can be useful to have some past experience with modeling (e.g., you can read a UML diagram or you have created some simple diagrams in the past), all the necessary steps will be explained in detail in chapter 3.

What is in this document?

This document provides a detailed tutorial on how to perform a LINDDUN PRO threat analysis of a system using a DFD[4] model of that system. An example is used to describe each of the steps in detail and assist you to get up and running in performing LINDDUN PRO threat analyses on your own software systems.

[4] DFD: Data Flow Diagram, a high-level description of your system in terms of processes, data stores, external entities, and the data flows between them. chapter 3 will provide more information on how to create this.

To get you started on your threat modeling journey, this tutorial first provides you an overview of the LINDDUN PRO threat modeling approach, the example used throughout the tutorial, and the material you need to perform the threat modeling.

Next, an overview and explanation of the LINDDUN threat types is provided as background, followed by the actual threat modeling steps in the following chapters.

When do I use this document?

Ideally, you start your privacy analyses as early as possible in the development lifecycle, so you can identify and address privacy threats early in the development lifecycle.

However, privacy threat modeling is not a single shot exercise that ends after one assessment. You should frequently re-perform analyses over time as the system evolves, new functionality gets introduced, old functionality removed, and new integrations with third party systems are added.

This ensures that you keep track of the most relevant privacy threats over time and that you won't miss any important threats that were introduced later on because the system changed over time.

1 Overview

1.1 LINDDUN PRO threat modeling approach

LINDDUN PRO is a systematic and in-depth approach to uncover privacy threats in a system. After creating a DFD of your system, you will iterate over every interaction between the DFD elements in the system to determine privacy threats in the sending, transfer, or receiving of personal data.

1.2 Required LINDDUN materials

To get started with LINDDUN PRO threat modeling, you will need the following materials (all available on the LINDDUN website):

Mapping table The mapping table outlines for every combination of DFD elements which LINDDUN threat types are applicable. It is included as table 4.1 in chapter 4.

Threat trees The threat trees provide a detailed breakdown of every LINDDUN threat type into its key characteristics (fig. 1.1). There are several variants of the trees available with different amounts of information, depending on your needs and familiarity with LINDDUN you can pick the variant with the level of information you need (basic; with examples; with criteria, impact, and additional information).

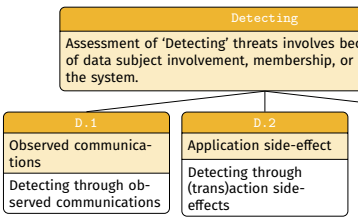


Figure 1.1: Partial detecting tree

1.3 Inputs needed of your analyzing your own system

In order to privacy threat model your system using LINDDUN PRO, a Data Flow Diagram or DFD describing your system is needed. If you already have such a diagram, you can skip the modeling (chapter 3) and go straight to the threat elicitation (chapter 4).

In case you still need to construct the DFD model of your system, you will require some inputs in order to be able to construct such a model. To create the model, you can either (1) rely on existing documentation, which can be internal textual documentation, requirements documents, design documents, or other types of diagrams; or (2) rely on the knowledge of software architects, designers, or programmers to assist you in constructing a high-level diagram of your system.

1.4 Running example of the tutorial

To illustrate the different LINDDUN PRO threat modeling steps, a running example will be used throughout this tutorial.

The system that will be used in this tutorial is that of a document processing service for automatically generating documents for organizations based on templates and tabular input data. This allows organizations to outsource the generation and delivery of, for example, monthly invoices by just providing the data and the relevant template to the service provider. The service provider will perform the generation and delivery of the documents through various channels such as email attachments, download links, or printed delivery through physical mail.

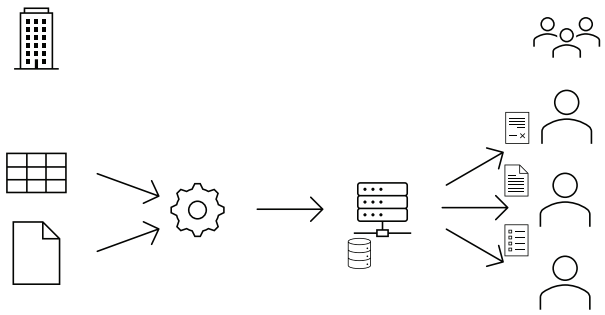


Figure 1.2: Schematic overview of the running example
This system processes tabular data and templates from organizations in order to automatically generate PDF invoices, pay slips, and other types of documents to distribute digitally and in print to different end-users.

Figure 1.2 provides a very high-level sketch of what this system looks like, depicting on the left-hand side, the organizations providing the tabular input and templates, and, on the right-hand side, the delivery of various generated documents to different types of users.

The following chapters will explain and illustrate the creation of the model and the privacy threat elicitation. This high-level diagram is a good illustration of how even a very basic sketch of the system can be a useful input in the following threat modeling steps.

2 LINDDUN Privacy Threat Types

2.1 Linking

Linking refers to any act of associating different data elements to each other (incl. meta-data) in such a way that it leads to undesirable privacy implications, i.e. when the combination of related data item will reveal (additional) information about a data subject (or groups of data subjects). By matching together several data items based on recurring attributes or properties, a user profile (or group profile) can be built. Simply put, **linking means learning more about an individual (or a group) by matching related data items together.** Linking typically relies on a recurring identifier, a combination of attributes (quasi-identifiers) or profile that allows data to be singled out. This means that one can be confident it all belongs to the same individual (without necessarily revealing the individual's identity). In addition, linking can also be applied to data of several individuals by matching similar properties in order to learn additional information about the group as a whole.

Many systems may require linking data items together to meet functional requirements (e.g., keep track of a user's session so they do not need to provide their login credentials for every request). However, 'Linking' threat analysis looks at situations in which this ability to tie things together, to learn or infer additional properties is considered problematic or undesirable.

2.2 Identifying

Personal data is by definition related to a data subject. Many systems may require identification of data subjects to meet their system goals. 'Identifying' threats however **express situations in which the identity of the data subject can be learned through leaks, can be deducted, or inferred when this is unwanted and to be prevented**.

2.3 Non-repudiation

Non-repudiation threats represent outcomes **in which an individual is not able to deny certain claims** specifically about their involvement in the system, or more broadly any claim pertaining themselves, as a consequence of the data collected, shared or an action taken by the individual (or other individuals) in the system.

Non-repudiation threats for a claim involve evidence with two dimensions: (i) the strength of that evidence with regard to the claim, (ii) the strength of the attributability to an individual.

2.4 Detecting

Assessment of 'Detecting' threats involves becoming aware of **data subject involvement, membership, or participation to the system** by observing existence of relevant information, through (i) observed communication, (ii) observed application side-effects (e.g., temporary files in the file system), or (iii) through system responses that may give away information about the existence of these elements. This final case addressed both threats caused by adversaries trying to probe the system (i.e., evoke responses that give away information about existence of data records), or system responses that accidentally leak information about the existence of specific records.

It is important to note that detecting threats do not require access to the data itself. Being able to observe the existence of stored data, side-effects of processes, or communication flows between certain parties can be sufficient to deduce additional relevant information about an individual.

2.5 Data Disclosure

A data disclosure is the transfer of personal data across a boundary, i.e. the collection of data by the system or the transfer of data to a known or unknown third party. From a privacy perspective, these disclosures should take into account the best interest of the involved data subjects. The minimality principle is key here. Only collect, process, store and share the strictly required personal data. **Data disclosure threats represent cases in which disclosure of personal data are considered *problematic*.** More precisely, 'Data Disclosure' threats represent cases in which either the *explicit* (i.e.

intended or designed) or the *implicit* (i.e. unintended or consequential) disclosure of personal data is considered *avoidable*.

When disclosures happen intentionally and *by design* (i.e. the system collects personal data to perform its functional goals) it is considered explicit. Conversely, disclosures are deemed implicit when they are indirect (i.e. through meta-data or derived from other data that is disclosed).

2.6 Unawareness and unintervenability

Unawareness and unintervenability focuses on the lack of support offered to the involved or affected individuals. Evaluating a system in terms of unawareness threats involves assessing the privacy harm caused by the system on a data subject by **insufficiently informing, involving, or empowering the data subject in its role and relation to the system**.

These threats capture three potential lacks of system support: (i) not properly informing data subjects about the collection of data and what is going on with that data in the system (lack of transparency), (ii) insufficiently making users that provide the system with personal data (about themselves or others) aware of the potential privacy harm or impact (lack of user feedback), and (iii) not providing data subjects with the required controls or means to influence how their data are being handled.

2.7 Non-compliance

Non-compliance is a general and broad notion that is defined as "**the lack of adherence to legislation, regulation, standards and best practices, leading to the incomplete management of risk**". Privacy-related risks should not be focused upon in a vacuum, and a privacy risk assessment is ideally complemented with broader attention to broader risk perspectives, such as legal risk, and cyber security risk.

Non-compliance as a LINDDUN threat type focuses predominantly on the intersection between the privacy threats and risks identified in the other threat types and the link to other risk notions, in both directions.

For example, documenting a 'Non-compliance threat' involves evaluating and articulating compliance-related problems that directly

derive from the applicable *Linking*, *Identifying*, *Non-repudiation*, *Data disclosure*, and *Unawareness* privacy threats, identifying security risks and translating these to broader perspectives of risk management (legal risk, cyber security risk, organizational risk).

3 Creating the DFD model

A common abstraction used in the context of threat modeling[1] is the Data Flow Diagram[2] or DFD. These diagrams provide a high-level overview of the system, the processes it contains, the external entities it interacts with, the data stores, and how data flows between these different elements. These types of diagrams provide

[1] Adam Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, Indianapolis, Indiana, 2014. ISBN 978-1-118-80999-0

[2] Tom DeMarco. *Structured Analysis and System Specification*. Yourdon Press, 1979

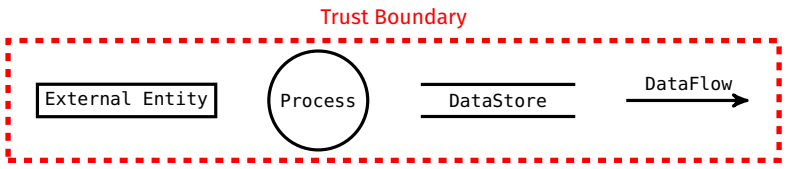


Figure 3.1: Overview of the types of elements in a Data Flow Diagram (DFD)

information on the data types (depending on the verbosity of the data flow labels), the processing and storage, and the disclosures to third parties (modeled as external entities). Figure 3.1 shows how the DFD elements are visualized.[3] A more detailed description of the element types follows below:

[3] There are other similar DFD notations, but in this context we use the DeMarco notation.

External entity (EE) represents any entity external to the system, these are users but also external services with which the system interacts. Both humans and systems can be external entities.

Process (P) represents any type of processing by the system.

Datastore (DS) represents any type of storage in the system (ranging from temporary files, local storage, in-memory, databases).[4]

[4] Note that any type of processing of, for example, database queries should be represented by process.

Dataflow (DF) represents a flow of data between the other elements.

Trust boundary (TB) can have multiple definitions. They can represent places where parties of different privilege levels interact, network or deployment boundaries, where security controls are enforced, etc. These are optional. If you use them, it is best to indicate their meaning.

Running example

To illustrate the construction of a DFD, we will start with the creation of a context DFD for the document processing running example. Afterwards, a more detailed DFD is created to model part of its functionality in-depth and to use for the later threat elicitation steps.

Creating the context DFD

Based on the high-level description of the running example given in section 1.4, an initial high-level context DFD diagram can be constructed. To construct this diagram, consider the system as a large single process, and write down all the different users and services it interacts with. If we perform this exercise for the running example, we get the following elements:

- A single system process to capture our entire document processing service.
- The customer organizations interact with this system to provide the raw data and templates for the document generation and delivery.
- The system interacts with the print service for the physical delivery of documents.
- The system interacts with banking services to provide digital delivery of invoices.
- The system interacts with email providers for the PDF delivery via email.
- End-users interact with the system to retrieve documents in their personal document store.

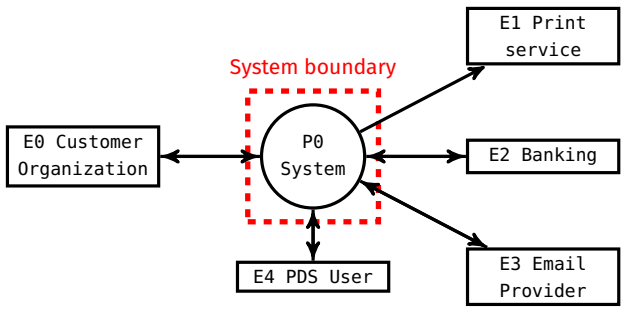


Figure 3.2: DFD context diagram the document processing and delivery service.

This gives you a very high-level overview of the system and its interactions with external components, after which you can start

breaking down the system process in order to add the more detailed inner workings to your model.

Creating the primary DFD

Next, we will create a more detailed primary DFD of the document processing and delivery service. This involves decomposing the system process from our context diagram in the previous section in order to reveal details about the underlying processes. When decomposing, you will need to keep track of the data flows on the previous diagram and make sure that every data flow on the original context diagram still has a destination or source on the original diagram.

To reduce the complexity of the diagram, we will ignore the interaction with the customer organization and how the documents will be generated. Instead, we will only focus on the scheduling and delivery of the generated documents.

When breaking down the system for scheduling and delivery will render the following elements:

- A scheduler process keeps track of generated documents and schedules their delivery.
- The delivery process interacts with the third parties to deliver the documents.
- A personal document store makes documents available to end users with an account on the service.

To store the relevant data, there is:

- An archive for storing the generated documents.
- Storage for the documents of the personal document store users.
- User data store for the personal document store service.

Figure 3.3 shows the final diagram all linked up the necessary data flows between the aforementioned elements. Note that the data flows to the external entities correspond with those on the context diagram (one-way to the print service and two-way to the other external entities) and that not all external entities interact with the same internal processes in our service.

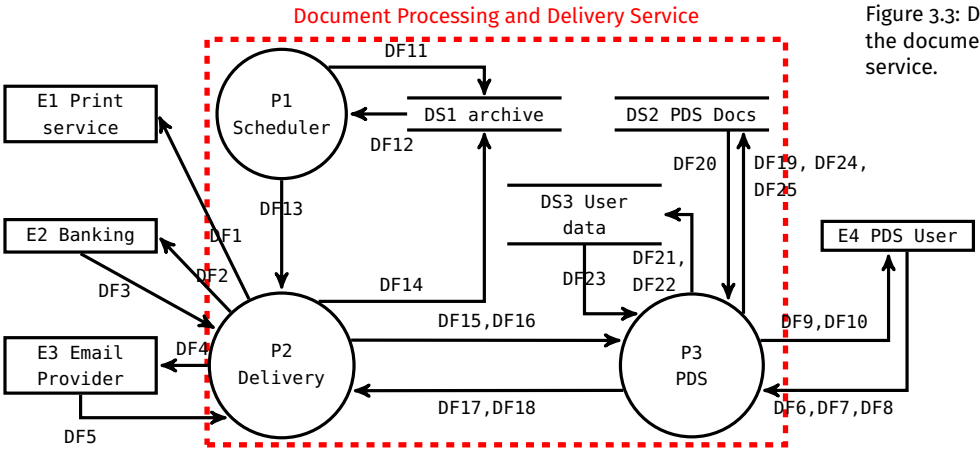


Figure 3.3: Data Flow Diagram (DFD) of the document processing and delivery service.

With this diagram at hand, we can move on to the next step and start eliciting privacy threats.

Notes on the simplifications in this diagram

To keep this diagram simple, the data flows are just numbered. On your own system, you would of course include information about the actual data that is transferred on the data flow labels. When we elicit threats on this diagram, the additional details on the data flow will be provided to show you how you use this information when eliciting privacy threats.

You probably also notice that some data flows have two labels. These are in fact two separate data flows. To avoid cluttering the diagram with many additional data flows, they are overlaid on top of one another with the two labels combined. You could do so yourself too, as long as it is clear whether your commas (or other separators) denote separate flows or denote multiple data elements as part of a single data flow.

4 Eliciting threats

After the construction of the DFD of the system under analysis, the next step involves the elicitation of privacy threats. This elicitation step involves systematically iterating over all the elements in the model to determine the applicability of the LINDDUN threats.

LINDDUN PRO uses interaction-based threat elicitation, where you iterate over every interaction[1] in our system model. Simply put, iterate over all the data flows in the model, and then consider whether there is a privacy threat at the source, at the data flow itself, or at the destination element.

[1] This is every combination of *source—dataflow—destination*.

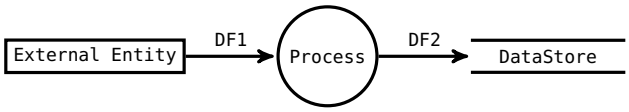


Figure 4.1: Example for interaction-based threat elicitation

For example, in the DFD in fig. 4.1 this involves iterating over the data flows*DF1* and *DF2*.

DF1 Assess the following locations for privacy threats.

- Assess whether there is a threat at the external entity sending data.
- Assess whether there is a threat at *DF1* itself.
- Assess whether there is a threat at the process receiving the data.

DF2 Similarly as for the previous data flow.

- Assess whether there is a threat at the process sending data.
- Assess whether there is a threat at *DF2* itself.
- Assess whether there is a threat at the data store receiving the data.

To help you in determining whether there is a threat at a particular location, you can use the following interpretations to decide whether there is a threat at the *source*, the *data flow*, or the *destination*:

Source The threat arises at the level of the element that shares or communicates data where the sharing of the data can cause a privacy threat. This is an **action**-effect threat as the source was triggered to initiate communication with the destination (e.g., a browser that retransmits cookies or other linkable identifiers to each recipient).

Data Flow The threat arises at the level of the data flow, i.e. when the data (both meta-data and the content itself) are in transit. These threats are **data**-centric (e.g., meta-data about the source and destination can be used to link multiple data flows, or to identify the parties involved in the communication).

Destination The threat arises at the level of the element that receives the data where the data can be processed or stored in a way that causes a privacy threat (e.g., insecure storage or insufficient minimization of the data upon storing). These threats are **action**-based as the receipt of the data and what the recipient does with that data triggers the threat.

Now that you know what to look for and how to interpret it, the next question is how to know if a particular LINDDUN threat is relevant. For this, you can refer to the mapping table (table 4.1).

SOURCE		DESTINATION	L	I	NR	D	DD	U	NC
Process	flow	Process	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D
Process		DataStore	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D
Process		ExternalEntity	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D
DataStore		Process	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D
ExternalEntity		Process	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D	S-fl-D

The elements in the columns highlight the element to which the privacy threats are associated (i.e. ***S***, ***fl***, ***D*** for source, flow, or destination respectively). Note that invalid DFD element combinations (such as *DataStore-flow-DataStore* or *ExternalEntity-flow-ExternalEntity*) are not included in this table.

Table 4.1: LINDDUN PRO Threat Type Mapping

For every interaction you encounter, you can use the mapping table to look up that interaction and check which LINDDUN threat types you need to consider. For the example diagram in fig. 4.1 you will need to check the last row[2] and the second row[3] to look up which LINDDUN threat types you need to consider.

[2] ExternalEntity—DataFlow—Process

[3] Process—DataFlow—DataStore

There are two main ways to start your iteration over your model and analyze the interactions to find privacy threats: starting with the threat types or starting from your model.

Starting from the threat types. When you start from the threat types, your iteration will start from a LINDDUN threat type and then go over your model elements. Hence, you start with *Linking*

then go over all the interactions in your model. After the *Linking* analysis, you move on to *Identifying* and then again go over all the interactions in your model. The benefit of this approach is that you remain entirely focused on a single threat type (and its tree) when doing the analysis. This can be faster, as you will probably remember most of the tree while you continue to go over your model elements. The downside is that you will go over your model multiple times.

Starting from the model. Another approach is to start from the model. This way, you will iterate over all the interactions in your model, and for each interaction review all of the LINDDUN threat types. You then move on to the next interaction and repeat the process. The benefit of this approach is that you do not have to revisit any element in the model, you will only go over your model once. The downside is that it can be slower or more difficult to switch between the different LINDDUN threat types for every interaction in your model.

Running Example

For the running example (fig. 1.2), we will select a number of data flows to illustrate how to elicit a number of different LINDDUN threat types using the LINDDUN PRO threat trees.[4]

[4] In this document, we will use partial trees without examples to keep the size of the trees small, but the LINDDUN website contains more versions of the trees with additional information to help you.

In the running example, we will not be analyzing all the data flows as this would be too lengthy to illustrate the mechanism. Rather, we will (1) pick a number of data flows from the example diagram, (2) provide you with a detailed description of the data flow label, (3) select one of the LINDDUN threat types, and (4) illustrate how to use the threat trees to come up with the privacy threats.

DF5 delivery status + doc ID (E3 Email Provider to P2 Delivery)

After delivery of the email with either a PDF attachment or URL to download the document, the email provider will return a delivery status for the provided document ID. This allows the document processing and delivery service to keep track of the successful delivery of the emails.

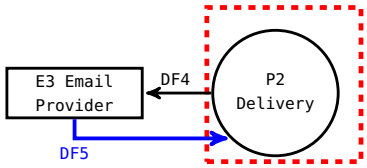


Figure 4.2: DFD focus on DF5

In this example, we will be considering the presence of *Linking* threats in this data flow. Next, we look at the mapping table (table 4.1) to determine if we need to consider the source, data flow,

or destination. Checking row number 5 (ExternalEntity—DataFlow—Process), we see that we need to consider all of them. We will thus look at the three of them, together with the *Linking* tree (fig. 4.3) to help us to elicit threats.

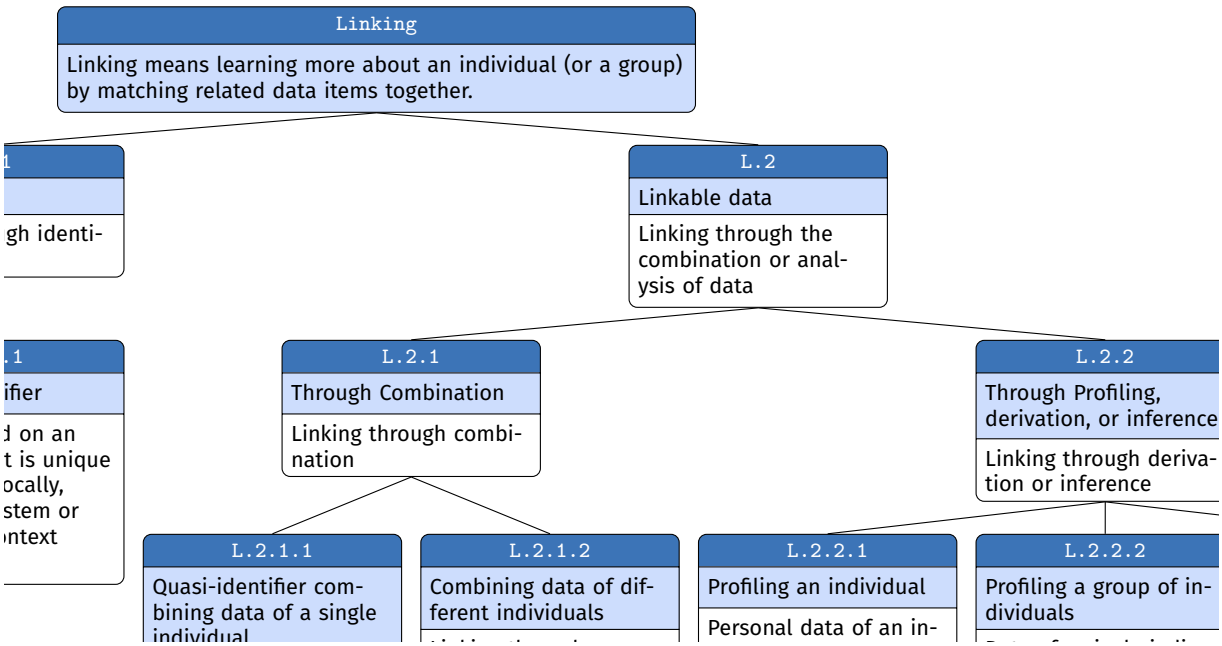


Figure 4.3: Linking tree

Source Since it is the email provider who notifies the system of a successful email delivery, there is not much to learn about this notification action, as none of the characteristics in the linking tree on identifiers or profiling allow us to learn anything about this notification action.[5] So for this interaction, we can skip *linking* at the source.

[5] This would be different for an end-user, as requests from users' browsers could trigger the transmission of cookies or other identifiers.

Data flow Next, we consider the data flow. On the data flow, the adversarial perspective becomes relevant, as in addition to the data on the data flow, there is also a lot of meta-data associated with the communication that an adversary could also use for profiling or linking. In this case, we'll look at `L.2.2.1` and `L.2.2.2`. For email domains with very few users[6], an adversary could profile users by linking document delivery confirmations with additional information on the customer organizations (for example, if the customer organizations or hospitals that use the service to send their invoices).

[6] For example, when someone hosts their own email.

Destination Finally, for the destination, we will again consider what our service does or could do with the data it receives. Here, we can again use `L.2.2.2` to conceive a threat where we try profile users by determining how well we can deliver documents to particular users. We could incorrectly infer that users with email are very responsive because delivery always succeeds almost instantly, while users with delivery via the document store are less responsive because we measure the

time they actually open document. This is, of course, a more far-fetched threat, but it could become more important if the delivery service would keep very detailed metrics of various delivery records.

A brief recap of our analysis of this interaction: we discovered two threats, one on the data flow (DF5) and one on the destination process (P2). Table 4.2 shows the overview of these threats. In the next chapter (chapter 5) we will discuss the more detailed documentation of these threats.

S	Fl	R	Location	Characteristics	Description
E3	DF5	P2	DF5	L.2.2.1, L.2.2.2	Adversary can profile users
E3	DF5	P2	P2	L.2.2.2	System profiles users

Table 4.2: Linking threats

After analyzing this data flow, we can move on to the next flow. The next flow will be between the same elements, but in the opposite directions. Furthermore, we will change the threat type and look for *Data Disclosure* threats.

DF4 PDF or URL with delivery info (P2 Delivery to E3 Email Provider)

This data flow represents the delivery of the email with either a PDF attachment or a URL to download the document, depending on the preferences set by the customer organization for the delivery of the document.

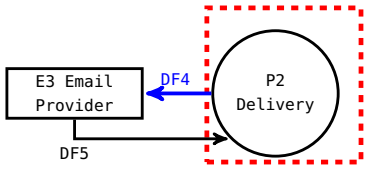


Figure 4.4: DFD focus on DF4

In this example, we will be considering the presence of *Data Disclosure* threats in this data flow. Next, we look at the mapping table (table 4.1) to determine if we need to consider the source, data flow, or destination. Checking row number 4 (Process—DataFlow—ExternalEntity), we see that we need to consider all of them. We will thus look at the three of them, together with the *Data Disclosure* tree (fragment shown in fig. 4.5) to help us.

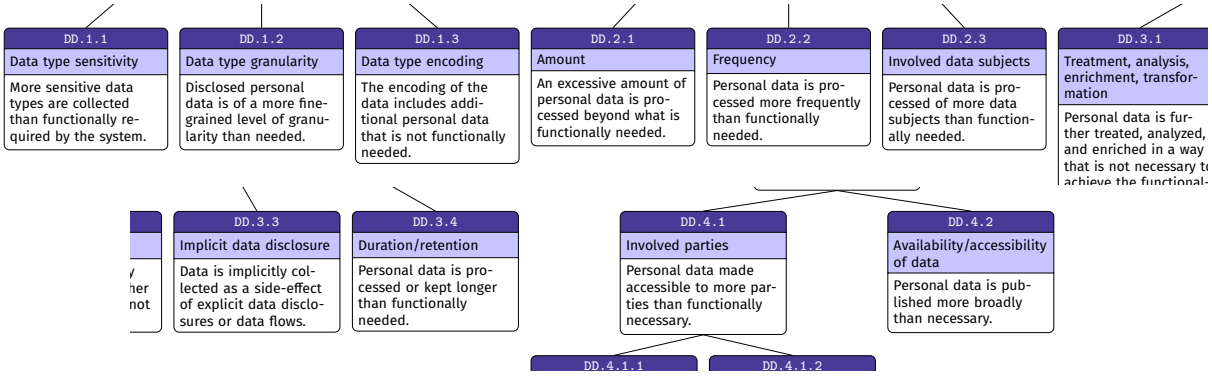


Figure 4.5: Fragments of the Data Disclosure tree

Source For the source, we again consider the potential threats because of the action our system takes. Looking at DD.1.1[7] and DD.4.1,[8] we could consider that we may be sending data that is much too sensitive to the third-party email provider (for example, if the documents contain health, financial, or insurance information). The email provider should not have access to this type of personal data.[9]

[7] on data sensitivity

[8] on the involved parties

[9] Assuming the data is not encrypted.

Data flow For the data flow, we look at the transfer of the information itself. Since email does not guarantee any kind of encryption,[10] the transfer of this data implicitly discloses (DD.3.3) it to all parties involved in the transfer of that information. An adversary or any (not necessarily malicious) user on the network could see these messages and learn sensitive data about the addressees.

[10] Note how we make an assumption or consider the absence of a countermeasure. This can be useful to document as an assumption.

Destination Finally, we look at the destination to consider the presence of Data Disclosure threats. When we look at DD.3.1, we consider that the email provider could analyze the contents of the different email messages addressed to the user to build up advertising profiles for the users. This would be a form of unnecessary processing of personal data.[11]

[11] Typically, this would be a more relevant consideration when the system you are building is performing these types of analyses or treatments for the data it receives, as there are limited options for you to prevent others from doing this.

Let us revisit our earlier table (table 4.2) and extend it with the threats we have found in this step (table 4.3).

S	Fl	R	Location	Characteristics	Description
E3	DF5	P2	DF5	L.2.2.1, L.2.2.2	Adversary can profile users
E3	DF5	P2	P2	L.2.2.2	System profiles users
P2	DF4	E3	P2	DD.1.1, DD.4.1	Excessively sensitive data shared
P2	DF4	E3	DF4	DD.3.3	Implicit disclosure to network users
P2	DF4	E3	e3	DD.3.1	Unnecessary analysis of data

Table 4.3: Adding the Data Disclosure threats

Next, we will look at the other side of the system, the communication between end users and the personal document store. Again, we will switch the threat type to add some variation.

DF7 URL (E4 PDS User to P3 PDS)

The data flow represents the user clicking on a customized URL in a received email in order to access a particular PDF document that was sent to them.

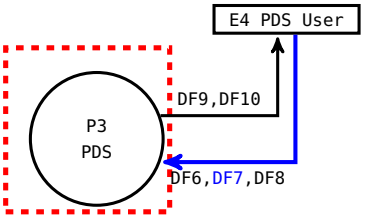


Figure 4.6: DFD focus on DF7

In this example, we will be considering the presence of *Detecting* threats in this data flow. Next, we look at the mapping table (table 4.1) to determine if we need to consider the source, data flow,

or destination. Checking row number 5 (ExternalEntity—DataFlow—Process), we see that we need to consider only the source and the data flow. We will thus look at those, together with the *Detecting* tree (fig. 4.7) to help us.

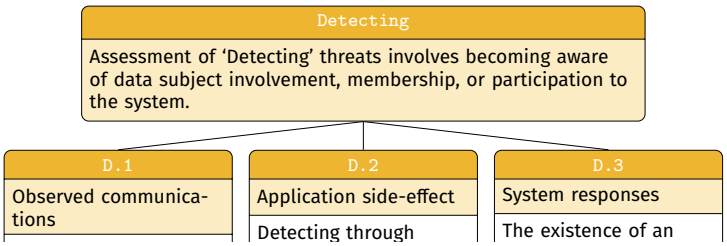


Figure 4.7: Detecting tree

Source For threats at the source, we look again at the *Detecting* in which `D.2` and `D.3` are the most relevant. Looking at `D.2`, we consider that many software applications might proactively scan URLs to detect and block malicious ones. In this case, the presence of the url in, for example, the user's mailbox could trigger the scan and thus, as a side-effect, send the request to the personal document store for retrieving the document. This would make it detectable that the user has received the URL.[12]

Data flow Looking at the data flow, `D.1` is highly relevant. An adversary could observe our requests to the personal document store. Even with all the communication encrypted, they could still detect that we have received a document.[13]

[12] This not only happens for security purposes. Many messaging applications follow URLs in chat messages to retrieve a preview image to include with the chat message.

[13] While this may not be a critical threat in the context of our document processing and delivery service, they could be much more important in voting systems, contact tracing systems, etc.

Finally, we revisit our earlier table (table 4.3), and further complete it with the detecting threats we have uncovered for this final interaction.

S	Fl	R	Location	Characteristics	Description
E3	DF5	P2	DF5	`L.2.2.1, L.2.2.2`	Adversary can profile users
E3	DF5	P2	P2	`L.2.2.2`	System profiles users
P2	DF4	E3	P2	`DD.1.1, DD.4.1`	Excessively sensitive data shared
P2	DF4	E3	DF4	`DD.3.3`	Implicit disclosure to network users
P2	DF4	E3	E3	`DD.3.1`	Unnecessary analysis of data
E4	DF7	P3	E4	`D.2`	Receiving a document URL can be detected
E4	DF7	P3	DF7	`D.1`	The retrieval of documents can be detected

Table 4.4: Adding the Detecting threats

In the next chapter, we look in more detail at how to elaborately document the threats and how to prioritize them.

5 Documenting and prioritizing threats

While this chapter discusses the steps of documenting and prioritizing threats separately from the elicitation in the previous chapter (chapter 4), these steps can usually be combined at the same time to improve the efficiency of your threat modeling exercise. However, to reduce the complexity of the explanation, they are discussed separately in this tutorial as this allows us to focus exclusively on a single aspect.

5.1 Documenting your threat model

Before delving into the details of documenting individual threats, we will first elaborate on the properties that are useful to document about the threat modeling exercise in general.[1,2]

[1] This information is also recorded in many threat modeling tools.

[2] Microsoft Corporation. Microsoft Threat Modeling Tool 7. http://aka.ms/tmt, 2022; OWASP. OWASP Threat Dragon. https://www.owasp.org/index.php/OWASP_Threat_Dragon, 2023; ThreatSpec. ThreatSpec. https://threatspec.org/, 2023; Izar Tarandach. PyTM. https://github.com/izar/pytm, 2022; and Schneider, Christian. Threagile. https://threagile.io/, 2021

Threat Model name The name of the threat model or the application being analyzed.

Description A brief description of the threat model.

Owner, contributors, and reviews The main owner(s), contributor(s), and reviewer(s) of the threat model.

High-level description A high-level description of the system that is threat modeled. This corresponds with the description of the running example provided in section 1.4.

Date The date the threat model was created and last modified.

LINDDUN version If you update or switch between threat knowledge bases, it can be useful to record the particular version that was used. This ensures that you can correctly interpret any external references to the threat knowledge.

Other documents Other documentation such as diagram and requirements documents that may be relevant for readers of the threat model.

Running example

When we briefly fill in these details, we get the following result for our document processing and delivery service (fig. 5.1)

Running example threat model	
Name	Document processing and delivery service
Description	This model describes the privacy threat analysis of the delivery services.
High-level description	This system automatically generates and delivers documents, allowing organizations to outsource the generation and delivery of documents such as invoices and pay slips.
Date	April 19, 2023
LINDDUN version	v1
Other documents	NA

Figure 5.1: Threat model description for the document processing and delivery service

5.2 Documenting your threats

Threat nbr A threat number or ID for later reference.

Title The name of threat.

Summary A more detailed summary describing the privacy threats.

DFD Elements The DFD elements that are involved in the threat. This includes the elements of the interaction (source, data flow, destination) and where the threat is located.

Threat Type The LINDDUN threat type of this threat.

Optionally, you could add the following information for a more detailed description of the threats.

Tree nodes The threat tree nodes you used to elicit the threat.

Assets involved Any assets that are involved or affected by the threat. For example, the types of personal data or affected users. This information can help you in determining the priority of the threat.

Priority The priority and its rationale. There are both categorical (high, medium, or low) and numerical risk analysis approaches that you could use to assess the priority of your threats.

Related threats Link to other relevant threats that could be addressed together, that could be caused by or cause this threat, or that could be informative in understanding this threat.

Comments Any other comments that could be useful to record.

Assumptions If you made any assumptions when eliciting the threats, it is highly recommended to document these with the threats as well. While you can document all the relevant assumptions as part of the individual threats, it may be useful to record them separately (section 5.3), so you can refer to the same assumptions multiple times.[3]

[3] Having a separate list can also help you to easily verify that you don't have any conflicting assumptions across multiple threats.

Running example

For creating more detailed documentation of the privacy threats in our document generation and delivery service, we will use our final table of threats (table 4.4) and put these into the more detailed template outlined above.

Running example threat description	
Nbr	T3
Title	Excessively sensitive data shared
Summary	Customer organizations can provide very sensitive documents to generate and deliver (for example, health, financial, or insurance). These types of sensitive personal should not be shared with third parties.
DFD Elements	***P2 Delivery***, *DF4, E3 Email provider*
Threat Type	Data Disclosure
Tree nodes	`DD.1.1, DD.4.1`
Assets involved	Sensitive personal documents (for example, medical bills or insurance documents)
Priority	High: improper sharing of personal data has legal implications, and the reputational damage for health and insurance customer organizations would be considerable.
Related threats	Relates to $T_{example}$: the customer organization is unaware that the selection of the document delivery channel influences the third parties with whom the personal data is shared.
Comments	We should add a mechanism for customer organizations to mark these documents as sensitive to prevent the delivery via insecure channels such as email.
Assumptions	A1 Assuming the PDF documents are not encrypted.

Figure 5.2: Threat description for the running example

5.3 Documenting your assumptions

The final element of documentation is the set of assumptions that were made during the threat modeling. Ideally, you record this set in parallel while documenting your threats. Assumptions do not have a lot of properties to record.

Assumption nbr The number or ID of the assumption so you can easily refer to the same assumption in multiple threats.

Assumption description The assumption itself.

Type Optionally, you could also record the type of the assumption. For example, whether it is an assumption about the system design, the presence (or lack) of certain countermeasures, the potential adversaries, etc.

5.4 Prioritizing your threats

The final element is the prioritization of the privacy threats you elicited. There are multiple ways you could prioritize your threats. LINDDUN PRO does not prescribe a particular method to use so you can use any existing method that best suits your needs. Below, we will give an overview of some of the prioritization approaches you could apply.

For the methods below, the first two approaches handle prioritization during the elicitation itself, while the last two approaches can be applied during the elicitation or afterwards on the entire list of privacy threats.

5.4.1 Selective analysis of your model

One way to prioritize up front and reduce the effort of your threat modeling exercise, is to be selective in which parts of your DFD model you will analyze for privacy threats.

For example, you could decide to focus only on the interactions with particular external entities, or only consider the data flows that cross the trust boundaries in your model.

An important sidenote to consider when using this method is that you cannot easily change this afterwards. If you decide to elicit threats for certain interactions only, reassessing the priority of the threats you have will not re-prioritize threats on interactions you never considered. In that case, you will have to do the threat elicitation on those other interactions as well.

5.4.2 *Filtering during the elicitation*

Another faster method is to do a quick assessment during the elicitation itself to determine whether the threat is important enough to record. Again, this method can save you some effort by not further delving into to details and documenting threats that you consider highly unlikely.

However, just as with the previous method, this approach does not allow you to reconsider that assessment. If you did not document the threat, you cannot update your assessment afterwards.

5.4.3 *Qualitative risk assessment*

This risk assessment method is the easiest to perform, as it does not involve any kind of numerical calculation. In this assessment, you consider the likelihood and impact on an ordinal scale[4] and combine those two assessments to provide an overall classification. This assessment is easier to perform but it can be more ambiguous as a lot of different situations would be grouped together in classification as medium. This can make reassessment afterwards more difficult as you do not really have the information that was at the basis of making the classification into a particular category.

[4] For example, low, medium, or high.

5.4.4 *Quantitative risk assessment*

The final approach involves a numerical risk assessment of threats. In this case you will assess the likelihood numerically as a frequency (for example, X incidents per year) as well as the impact. Assessing the impact numerical will require you to establish some unit to measure the damage of the threats.[5] This approach is usually a lot more heavyweight, as it requires you to specify all the input values for the risk assessment. The benefit is that automation can be used to reduce this effort and to make reassessments if the inputs change trivial to recalculate.

[5] This could be a financial value or a unitless damage value.

Bibliography

Ann Cavoukian. Privacy by design [leading edge]. *IEEE Technology and Society Magazine*, 31(4):18–19, 2012. DOI: 10.1109/MTS.2012.2225459.

Tom DeMarco. *Structured Analysis and System Specification*. Yourdon Press, 1979.

European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union*, 59(L 119):1–88, May 2016.

Microsoft Corporation. Microsoft Threat Modeling Tool 7. http://aka.ms/tmt, 2022.

OWASP. OWASP Threat Dragon. `https://www.owasp.org/index.php/OWASP_Threat_Dragon`, 2023.

Schneider, Christian. Threagile. `https://threagile.io/`, 2021.

Adam Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, Indianapolis, Indiana, 2014. ISBN 978-1-118-80999-0.

Izar Tarandach. PyTM. `https://github.com/izar/pytm`, 2022.

ThreatSpec. ThreatSpec. `https://threatspec.org/`, 2023.