Threat Modeling

Martin Stanek

2025

Table of Contents

Introduction

Terminology

Methodologies (STRIDE, PASTA, LINDDUN)

What is Threat Modeling

- systematic and structured process used to identify, analyze and document
  - potential threats and vulnerabilities
  - security requirements/issues that must be implemented/addressed
- threat model (TM) – the output of the threat modeling process
  - diagrams, tables, report, etc.
- primary use for software engineering
  - threat modeling aims at design flaws
- scale: single application, server, information system, infrastructures, networks, etc.
- other areas that use threat modeling:
  - risk management (assessment of threats that were, or not, considered)
  - cyber wargaming (TM defines scenarios)
  - security operations

Threat Modeling and SDLC

- ideal time to start when architecture is known
  - major components, functions, data flows
- threat modeling can be useful in any phase of the SDLC
  - late inclusion – more costly remediation (if needed)
  - later phases – more detailed TM (decide when enough is enough)
- revise TM
  - substantial changes are designed/implemented (new features, change in architecture)
  - security incident happens

Threat Modeling and Security Testing

Why do we care?

1. Threat modeling as a method to define scope for security testing
   - What should we focus on?
2. TM as an input for security testing:
   - Architecture description
   - What threats have been considered – is the TM complete?
   - What security requirements/controls are in place?
3. Threat modeling as a tool to analyze implemented/deployed system for design flaws.

Reality:

- threat modeling is rarely performed (time, effort vs. perceived benefits)

Terminology

- Threat, Vulnerability, Weakness, Risk
  - subtle differences among standardization documents, frameworks, tools, etc.
  - used much more broadly in practice
- definitions from the Glossary of key information security terms (NIST)

Threat

Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

- Examples:
  - (Spear) Phishing
  - Unauthorized access
  - Data leak
  - (Distributed) Denial of Service
  - Exploit recent vulnerabilities

Vulnerability and Weakness

Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

Weakness / Vulnerability examples:

- Improper Restriction of Operations within the Bounds of a Memory Buffer (CWE-119)
    - Heartbleed (CVE-2014-0160)
- Improper Input Validation (CWE-20)
    - Windows SMB v1 remote code execution (CVE-2017-0144) – EternalBlue exploit
- Authentication Bypass by Capture-replay (CWE-294)
    - Outlook NTLM Vulnerability (CVE-2023-23397)

Risk

A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.

- usually a qualitative measure:
    - NIST 800-30r1: Very Low, Low, Moderate, High, Very High
    - OWASP Risk Rating Methodology: Note, Low, Medium, High, Critical
    - likelihood and impact are measured qualitatively as well

Threat modeling automation

- Automation:
  - architecture (DSL, IaC, code) → diagrams and reports
- Threatspec (last update in 2019)
  - threat modeling annotations as comments in code, low-level modeling
- ThreatPlaybook (last update 2020)
  - "Story-Driven Threat Modeling", YAML file
- pytm (last update 2024)
  - a python program describes an application (system), own library of threats
- Threagile (last update 2024)
  - "agile" approach, architecture described in a YAML file
- GUI tools to create a TM:
  - OWASP Threat Dragon, Microsoft Threat Modeling Tool
  - usually follow/support established methodologies

Methodologies (STRIDE, PASTA, LINDDUN)

- Microsoft Threat Modeling Process (STRIDE inside)
- PASTA (Process for Attack Simulation and Threat Analysis)
- ATASM (Architecture, Threats, Attack Surfaces, Mitigations)
- TARA (Threat Assessment & Remediation Analysis)
- LINDDUN (privacy threat modeling)

Microsoft Threat Modeling Process / STRIDE

- software-centric modeling
  - other approaches: asset-centric, attacker-centric

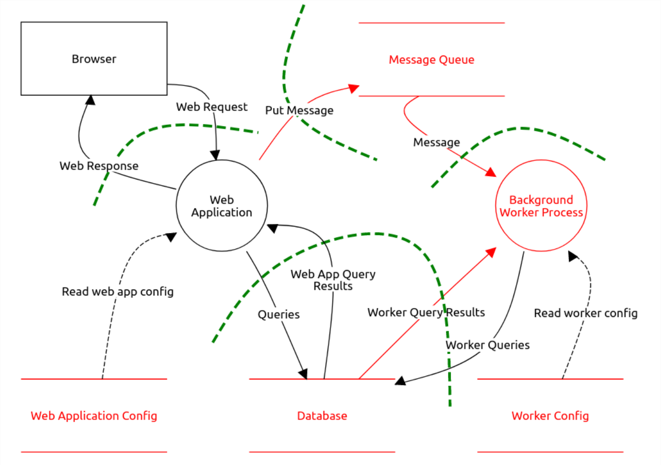
Process:

1. (Diagram) What are you building?
2. (Identify) What can go wrong with it once it's built?
3. (Mitigate) What should you do about those things that can go wrong?
4. (Validate) Did you do a decent job of analysis?

Diagram

- data flow diagrams (DFD)
  - other options: sequence diagrams, state diagrams
- elements of DFD:
  - process: program, library, web frontend, ...
  - data flow: HTTPS, RPC, e-mail, ...
  - data store: file share, database, shared memory, ...
  - external entity: user, organization, another system, ...
  - *trust boundary* (extension of DFD) – usually between elements communicating trough network
- complex systems: multiple diagrams, decomposition

DFD example



Sample model from OWASP Threat Dragon

- circle – process
- rectangle – actor
- arrow – data flow
- “double” line – store
- green line – trust boundary

- dashed arrow – out of scope
- red color – threats were identified

STRIDE – Identify

Threat	Property	Examples
Spoofing	Authentication	file, machine, person, role
Tampering	Integrity	file, memory, network
Repudiation	Non-Repudiation	action, log tampering
Information Disclosure	Confidentiality	eavesdropping, temp files, ACL
Denial of Service	Availability	network resources, storage, CPU
Elevation of Privilege	Authorization	errors in authorization

- identify relevant threats for elements in the scope
- security expertise vs. threat/attack trees, CAPEC, OWASP Top 10, etc.

STRIDE-per-Element

- simplified approach
- some threats are more relevant for particular elements
- focus on set of threats for each element
    - '?' – usually not relevant, except for logs that address repudiation threats
- might miss something, the table might be inaccurate for the system

Element	S	T	R	I	D	E
process	x	x	x	x	x	x
data flow		x		x	x	
data store		x	?	x	x	
external entity	x		x			

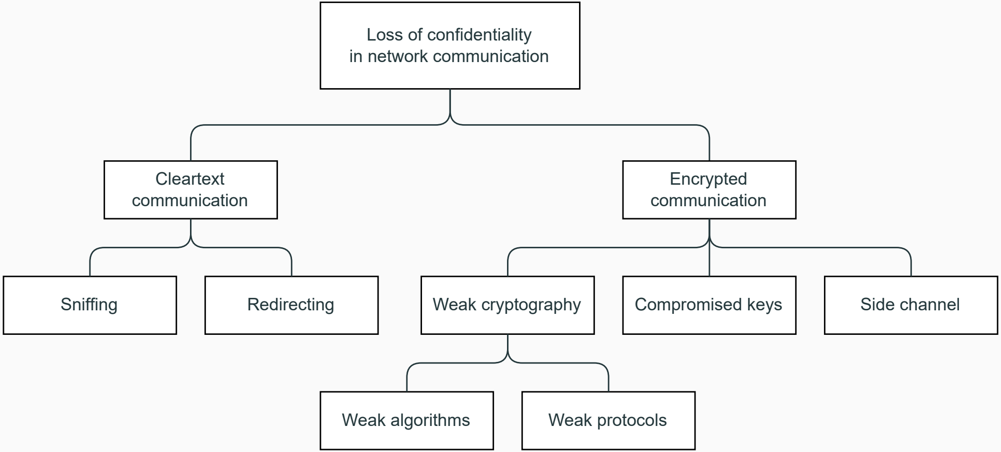
Other STRIDE Variants

- STRIDE-per-Interaction
    - (origin, destination, interaction)
    - identify interactions that cross a trust boundary
    - apply STRIDE to each interaction
- experimental comparison with STRIDE-per-element
    - just a few studies (inconclusive)
    - more complex, (slightly) more false-positives
    - less true positive threats per hour (it takes longer to find relevant threats)
- DESIST
    - Dispute, Elevation of privilege, Spoofing, Information disclosure, Service denial, Tampering
- STRIDE-LM (LM – Lateral Movement)

Attack Trees

- application of attack trees
  - find threats (alternative to STRIDE or other threat-finding methods)
  - organize threats found by other methods
- lack of ready-to-use attack trees
- creating a new attack tree is hard
  - brainstorming, expertise, attack libraries
  - completeness – how to ensure that relevant attack are not overlooked
  - scope – what attack are out of scope, e.g., hardware attacks when analyzing web application
  - level of detail – how granular the tree should be (where to stop)
- various graphical presentations

Attack Tree example



Loss of confidentiality in network communication

Cleartext communication

Encrypted communication

Sniffing

Redirecting

Weak cryptography

Compromised keys

Side channel

Weak algorithms

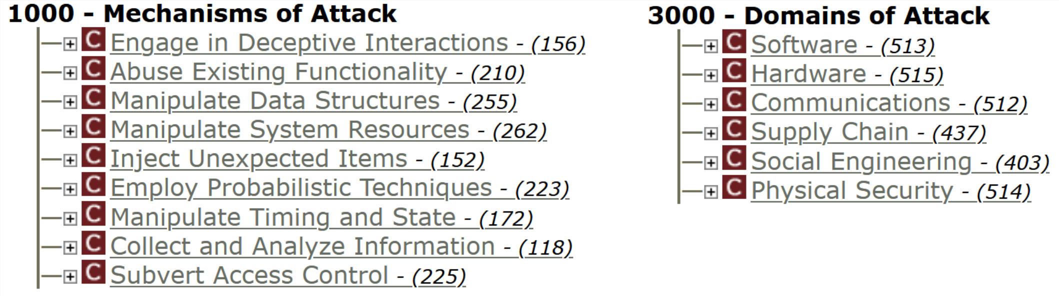
Weak protocols

Attack Libraries (CAPEC)

- MITRE CAPEC (Common Attack Pattern Enumeration and Classification)
  - structured catalog of common attack patterns (more than 550)
  - two main hierarchical views: mechanisms of attack, domains of attack
- pattern types / abstraction levels (hierarchy is not always complete):
  - Category – collection of attack patterns based on some common characteristic
  - Meta – abstract characterization of a specific methodology or technique
  - Standard – a specific methodology or technique
  - Detailed – a low level of detail, a specific technique and technology
  - example:

    (C) Subvert Access Control – (M) Authentication Bypass – (S) Escaping Virtualization –
    (D) Escaping a Sandbox by Calling Code in Another Language

CAPEC – Mechanisms of attack, Domains of attack



1000 - Mechanisms of Attack

- ⊞ C Engage in Deceptive Interactions - *(156)*
- ⊞ C Abuse Existing Functionality - *(210)*
- ⊞ C Manipulate Data Structures - *(255)*
- ⊞ C Manipulate System Resources - *(262)*
- ⊞ C Inject Unexpected Items - *(152)*
- ⊞ C Employ Probabilistic Techniques - *(223)*
- ⊞ C Manipulate Timing and State - *(172)*
- ⊞ C Collect and Analyze Information - *(118)*
- ⊞ C Subvert Access Control - *(225)*

3000 - Domains of Attack

- ⊞ C Software - *(513)*
- ⊞ C Hardware - *(515)*
- ⊞ C Communications - *(512)*
- ⊞ C Supply Chain - *(437)*
- ⊞ C Social Engineering - *(403)*
- ⊞ C Physical Security - *(514)*

CAPEC – Pattern Structure

depends on the pattern type, common fields for abstraction level *detailed*:

- Description
- Likelihood Of Attack
- Typical Severity
- Relationships
  - ChildOf, ParentOf, CanFollow, View Names/Top Level Categories
- Execution Flow
  - Explore, Experiment, Exploit
- Prerequisites
- Skills Levels
- Resources Required
- Indicators
- Consequences
- Mitigations
- Example Instances
- Related Weaknesses (CWE)
- References

Mitigate (1)

- address identified threats
- mitigations – dig deeper
    - attacker will try to bypass the mitigation/security control
    - examples:
        - encrypted communication → obtain encryption keys
        - password authentication → misuse password reset function
- track assumptions (and validate them)
    - describe: assumption, what if the assumption is wrong
    - examples:
        - attacker does not have physical access to the server
        - no feasible cryptanalytic attack on algorithms in use

Mitigate (2)

- include security notes:
  - how to use the system, API and application securely (considering implemented mitigations)
  - security considerations what is in/out of scope and why, residual risks, etc.
  - any other notes and observations from threat modeling

Mitigate – security controls

- not covered in this lecture
- compliance: policies, regulatory requirements
  - selection and implementation of controls should meet these requirements
- threat modeling – part of risk management
- examples of measures to address STRIDE threats:
  - Spoofing: authentication (data, computers, people) – digital signatures, TLS, 2FA
  - Tampering: integrity – access control, network protocols
  - Repudiation: non-repudiation – logging, digital signatures, hash trees, TTP
  - Information Disclosure: confidentiality – access control, encryption, key management
  - Denial of Service: availability – HA design, cloud, quotas
  - Elevation of Privilege: authorization – ACLs, runas/sudo, sandboxing

Validate

- subset of all test of a system
- threat model testing
    - the model matches the reality
    - all threats are addressed adequately
    - mitigation controls are tested for correctness and effectiveness
- significant overlap with security testing

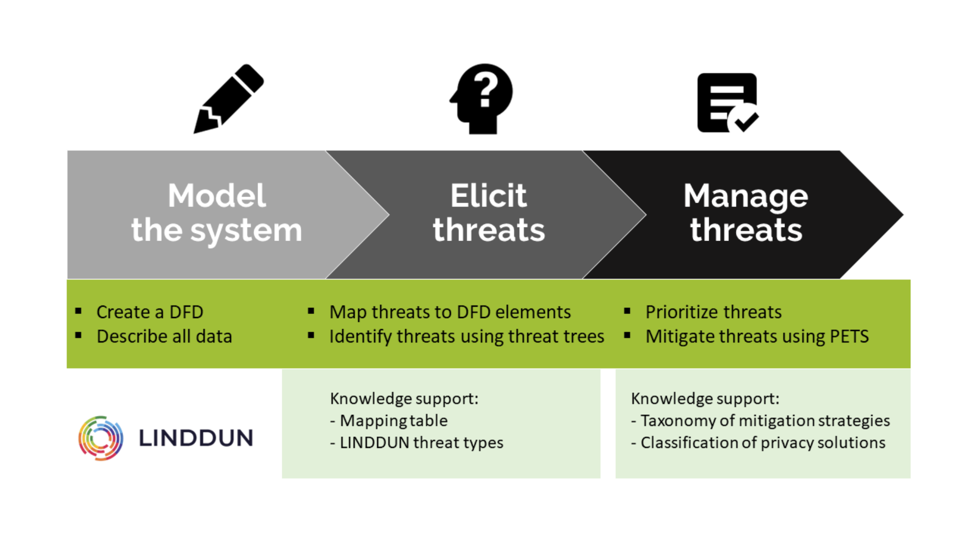
PASTA

- Process for Attack Simulation and Threat Analysis
- risk-centric approach
- comprehensive
  - 7 stages and $4 + 5 + 3 + 6 + 5 + 6 + 4 = 33$ activities
  - RACI matrix (Responsible, Accountable, Consulted, Informed) has 14 roles
    - Management, Business Analyst, Architecture, . . . , Security Operations, Threat Modeler
- stages
  1. Definition of the Objectives
  2. Definition of the Technical Scope
  3. Application Decomposition and Analysis
  4. Threat Analysis
  5. Weakness and Vulnerability Analysis
  6. Attack Modeling and Simulation
  7. Risk Analysis and Management

LINDDUN

- LINDDUN – privacy threat modeling methodology
- flavors:
  - LINDDUN GO (lean privacy analysis)
  - LINDDUN PRO (systematic privacy analysis)
  - LINDDUN MAESTRO (model-driven analysis, *not published yet*)
- differences
  - required privacy expertise (novice vs. expert)
  - primary input (sketch vs. DFD vs. enriched DFD)
  - effort (main focus on analysis vs. modeling)
  - analysis type (brainstorm vs. manual/tool assisted)

LINDDUN PRO steps



Model the system

- Create a DFD
- Describe all data

Elicit threats

- Map threats to DFD elements
- Identify threats using threat trees

Knowledge support:
- Mapping table
- LINDDUN threat types

Manage threats

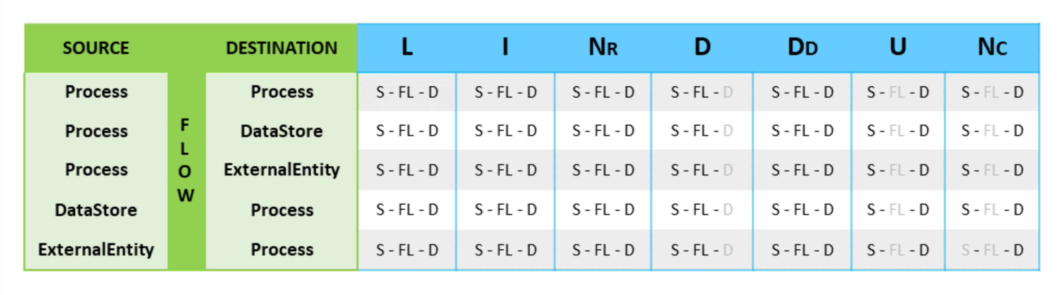
- Prioritize threats
- Mitigate threats using PETS

Knowledge support:
- Taxonomy of mitigation strategies
- Classification of privacy solutions

LINDDUN

LINDDUN PRO mapping

- per-interaction approach
  - identify all interactions: source + data flow + destination
- mapping table indicates what threats should be considered for each interaction type
- 7 threat types (*see next slides*), detailed breakdown in privacy threat trees



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

LINDDUN – Privacy Threats 1

- Linking
  - Learning more about an individual or a group by associating data items or user actions.
- Identifying
  - Learning the identity of an individual.
- Non-Repudiation
  - Being able to attribute a claim to an individual.
- Detecting
  - Deducing the involvement of an individual through observation.
- Data disclosure
  - Excessively collecting, storing, processing or sharing personal data.

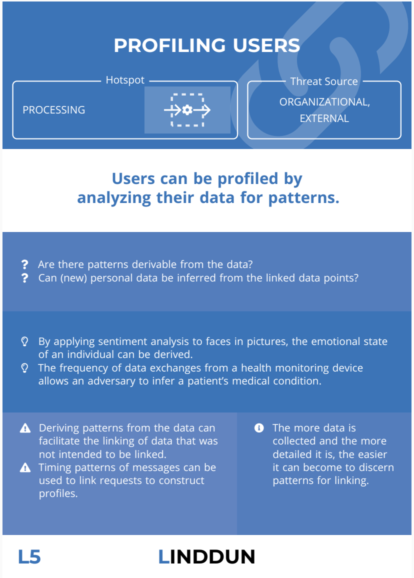
LINDDUN – Privacy Threats 2

- Unawareness, Unintervenability
  - Insufficiently informing, involving or empowering individuals in the processing of their personal data.
- Non-compliance
  - Deviating from security and data management best practices, standards and legislation.

LINDDUN GO

- simplified approach to LINDDUN, a card game
- quick start to privacy threat modeling
- beginners, brainstorming, 33 threat cards
- DFD not necessary – use anything that helps to identify *hotspots*
  - inbound flow, outbound flow, data storage, data retrieval, processing
  - additional info to hotspots: personal data
- three main threat sources are considered:
  - external attacker, organizational, receiving party

LINDDUN GO – sample threat type cards



PROFILING USERS

Hotspot: PROCESSING

Threat Source: ORGANIZATIONAL, EXTERNAL

Users can be profiled by analyzing their data for patterns.

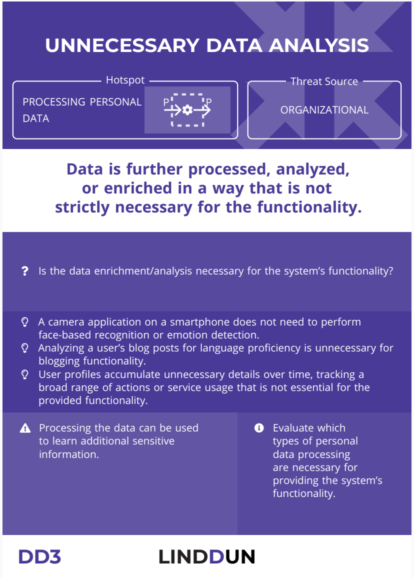
- ? Are there patterns derivable from the data?
- ? Can (new) personal data be inferred from the linked data points?

- 💡 By applying sentiment analysis to faces in pictures, the emotional state of an individual can be derived.
- 💡 The frequency of data exchanges from a health monitoring device allows an adversary to infer a patient's medical condition.

- ⚠ Deriving patterns from the data can facilitate the linking of data that was not intended to be linked.
- ⚠ Timing patterns of messages can be used to link requests to construct profiles.

- ℹ The more data is collected and the more detailed it is, the easier it can become to discern patterns for linking.

L5 LINDDUN



UNNECESSARY DATA ANALYSIS

Hotspot: PROCESSING PERSONAL DATA

Threat Source: ORGANIZATIONAL

Data is further processed, analyzed, or enriched in a way that is not strictly necessary for the functionality.

- ? Is the data enrichment/analysis necessary for the system's functionality?

- 💡 A camera application on a smartphone does not need to perform face-based recognition or emotion detection.
- 💡 Analyzing a user's blog posts for language proficiency is unnecessary for blogging functionality.
- 💡 User profiles accumulate unnecessary details over time, tracking a broad range of actions or service usage that is not essential for the provided functionality.

- ⚠ Processing the data can be used to learn additional sensitive information.

- ℹ Evaluate which types of personal data processing are necessary for providing the system's functionality.

DD3 LINDDUN

PLOT4ai

- PLOT4ai: Privacy Library Of Threats 4 Artificial Intelligence
- inspired by LINDDUN GO, similar card-based approach
- 8 categories (86 threats)
  1. Technique & Processes
  2. Accessibility
  3. Identifiability & Linkability
  4. Security
  5. Safety
  6. Unawareness
  7. Ethics & Human Rights
  8. Non-compliance

Exercises

- Choose one: Microsoft Threat Modeling Tool or OWASP Threat Dragon.
- Choose an application, *invent* details if necessary
- Perform a partial threat modeling.
- Send your model and be prepared to present and discuss its details.

1. *Create a DFD for your application, suitable for the STRIDE-per-element modeling.*
2. *Find and describe a threat for each threat type from the STRIDE. If the modeling tool generates threats automatically, identify the most and the least relevant ones.*
3. *Find and document two privacy threats (of different threat types) using LINDDUN GO.*

Additional Resources

1. Adam Shostack, *Threat Modeling: Designing for Security*, Wiley, 2014.
2. Tony UcedaVelez, Marco M. Morana, *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*, Wiley, 2015.
3. Deborah J. Bodeau et al., *Cyber Threat Modeling: Survey, Assessment, and Representative Framework*, HSSEDI, 2018.
4. Michael Muckin et al., *A Threat-Driven Approach to Cyber Security*, Lockheed Martin, 2019.