
SBA
Research

Threat Modeling Training

App Sec Training @ NÖ Landesregierung 2025-12-10

SBA Research

Federal Ministry Innovation, Mobility and Infrastructure Republic of Austria
Federal Ministry Economy, Energy and Tourism Republic of Austria
FFG Promoting Innovation.
vienna business agency
For the City of Vienna
European Commission
FWF Austrian Science Fund
netidee FÖRDERUNGEN

whoami

- Andreas Boll
- IT Security Consultant at SBA Research
  - Penetration testing
  - Source code audits
  - SDLC
  - Threat Modeling

aboll@sba-research.org

Class “Rules”

- What happens in this course, stays in this course!
- Please Participate!
  - Ask questions
  - Share experiences & opinions

Introduction

Why Threat Modeling?

Privacy Fail

NETZPOLITIK

Datenschützer üben scharfe Kritik an österreichischen Plänen für den grünen Pass

Nutzung von E-Card würde massenhafte Datensammlungen und Stalking ermöglichen – Zentrales Abfragesystem erlaube Überwachung der Aktivitäten und sei potenziell EU-rechtswidrig

Andreas Proschofsky 7. Mai 2021, 09:13 564 Postings

Zentraler Ansatz als Problem

Und auch für eine weitere Problematik macht dies wenig Unterschied: Die Verwendung eines zentralen Servers zur Abfrage des Covid-Status eröffne die Möglichkeit, dass hier exakt mitprotokolliert werden könnte, wer wann und wo Einlass begehrt. Theoretisch wäre es also möglich, in einem gewissen Maß Bewegungsprofile über die Nutzer zu erstellen. Nun wollen die Datenschützer den Betreibern solche Absichten nicht gleich unterstellen, äußern aber Unverständnis darüber, wie man auf solche Ideen komme. Immerhin hätten in den vergangenen

https://www.derstandard.at/story/2000126461757/datenschuetzer-ueben-scharfe-kritik-an-oesterreichischen-plaenen-fuer-den-gruenen

“Shift Left”

"*Weeks of coding can save you hours of planning*"

Secure Design vs. Secure Coding

Secure Design	Secure Coding
Requires secure design knowledge & contextual knowledge	Requires security knowledge from every developer
Yields security requirements	Secure coding profits from secure design
Manual task	Can be (partially) automated

SDLC Maturity

Model | Design | Threat Assessment

The Threat Assessment (TA) practice focuses on identifying and understanding of project-level risks based on the functionality of the software being developed and characteristics of the runtime environment. From details about threats and likely attacks against each project, the organization as a whole operates more effectively through better decisions about prioritization of initiatives for security. Additionally, decisions for risk acceptance are more informed, therefore better aligned to the business.

By starting with simple threat models and building application risk profiles, an organization improves over time. Ultimately, a sophisticated organization would maintain this information in a way that is tightly coupled to the compensating factors and pass-through risks from external entities. This provides greater breadth of understanding for potential downstream impacts from security issues while keeping a close watch on the organization's current performance against known threats.

Maturity level		Stream A Application Risk Profile	Stream B Threat Modeling
1	Best-effort identification of high-level threats to the organization and individual projects.	A basic assessment of the application risk is performed to understand likelihood and impact of an attack.	Perform best-effort, risk-based threat modeling using brainstorming and existing diagrams with simple threat checklists.
2	Standardization and enterprise-wide analysis of software-related threats within the organization.	Understand the risk for all applications in the organization by centralizing the risk profile inventory for stakeholders.	Standardize threat modeling training, processes, and tools to scale across the organization.
3	Proactive improvement of threat coverage throughout the organization.	Periodically review application risk profiles at regular intervals to ensure accuracy and reflect current state.	Continuously optimization and automation of your threat modeling methodology.

https://owaspsamm.org/model/design/threat-assessment/

Goal

Get a deep understanding of the system you are building and how to protect it.

Have a dependable lack of surprise

Benefits

- Team has a shared view on security
- Prevent security design flaws & get concise security requirements
- Greatest risks are known & addressed
- Risk based approach to fixing problems
- Cheap way of fixing/preventing vulnerabilities
- Helps you get funding for implementing security controls
- Security documentation as a byproduct

Possible Problems

- Need (external) expertise
- Time consuming (a few hours per month, initial: ~1 week)
- Hard to reproduce across different teams
  - Different approaches
- Subpar tool support
- Model gets outdated

Tool support

- Microsoft Threat Modeling Tool
- OWASP Threat Dragon (Alpha)
- IRIUS Risk (expensive)
- Threagile, pytm (open source)
  - Semantic statements within the source code
  - Hard for beginners

- Whiteboard + Lists (ad-hoc)

Basic Steps

1. What are we building?
    - Diagram
2. What can go wrong?
    - Find threats
3. What can we do about it?
    - Quantify risk
    - Define mitigations
    - Accept residual risks

Real Life Threat Modeling

Defend your house/apartment

- Value of what you are protecting?
- Threat actors?
- Cost of security controls?

More:
https://danielmiessler.com/blog/everyday-threat-modeling/

Organization

Duration

- Initial session: a few days
- Regular updates: about 1-2 hours per session (weekly, monthly)
  - Stay on time
- Prefer multiple short sessions over rare long ones

Duration

- Scope your sessions
  - Stick to your scoping!
  - Avoid to lose yourself in details
- Sessions can focus on many aspects
  - New features
  - More detailed explorations
  - Repeat past evaluations

Locally vs. Remote

Threat Modeling can be performed well both on-site and in a remote setting.

- On-site makes communication easier
- Use of remote communication tools can help speed up the “prettifying” part after the session for documentation purposes
- Getting the right people “in the room” might be easier in one of the scenarios
- Use of real vs. online whiteboards

Pitfalls

- Spending too much time
  - Loose yourself in details
  - Want to build the “perfect model”
  - Waste time on not relevant threat (“nation state”)
- Waiting too long to start
  - You don’t need to be perfect to start threat modeling
  - Use your first sessions as learning opportunity

Terminology

Vulnerability

A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

RFC 2828: https://datatracker.ietf.org/doc/html/rfc2828

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.

*That is, a threat is a **possible danger that might exploit a vulnerability**. A threat can be either "**intentional**" (i.e., intelligent; e.g., an individual cracker or a criminal organization) or "**accidental**" (e.g., the possibility of a computer malfunctioning, or the possibility of an "act of God" such as an earthquake, a fire, or a tornado).*

RFC 2828: https://datatracker.ietf.org/doc/html/rfc2828

Threat actor

An individual or a group posing a threat.

NIST Glossary: https://csrc.nist.gov/glossary/term/threat_actor

Sometimes also called a **threat agent.**

Threat modeling

A form of risk assessment that models aspects of the attack and defense sides of a logical entity, such as a piece of data, an application, a host, a system, or an environment.

NIST Glossary: https://csrc.nist.gov/glossary/term/threat_modeling

Risk

An expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular harmful result.

RFC 2828: https://datatracker.ietf.org/doc/html/rfc2828

Risk Assessment/Analysis

A process that systematically identifies valuable system resources and threats to those resources, quantifies loss exposures (i.e., loss potential) based on estimated frequencies and costs of occurrence, and (optionally) recommends how to allocate resources to countermeasures so as to minimize total exposure.

RFC 2828: https://datatracker.ietf.org/doc/html/rfc2828

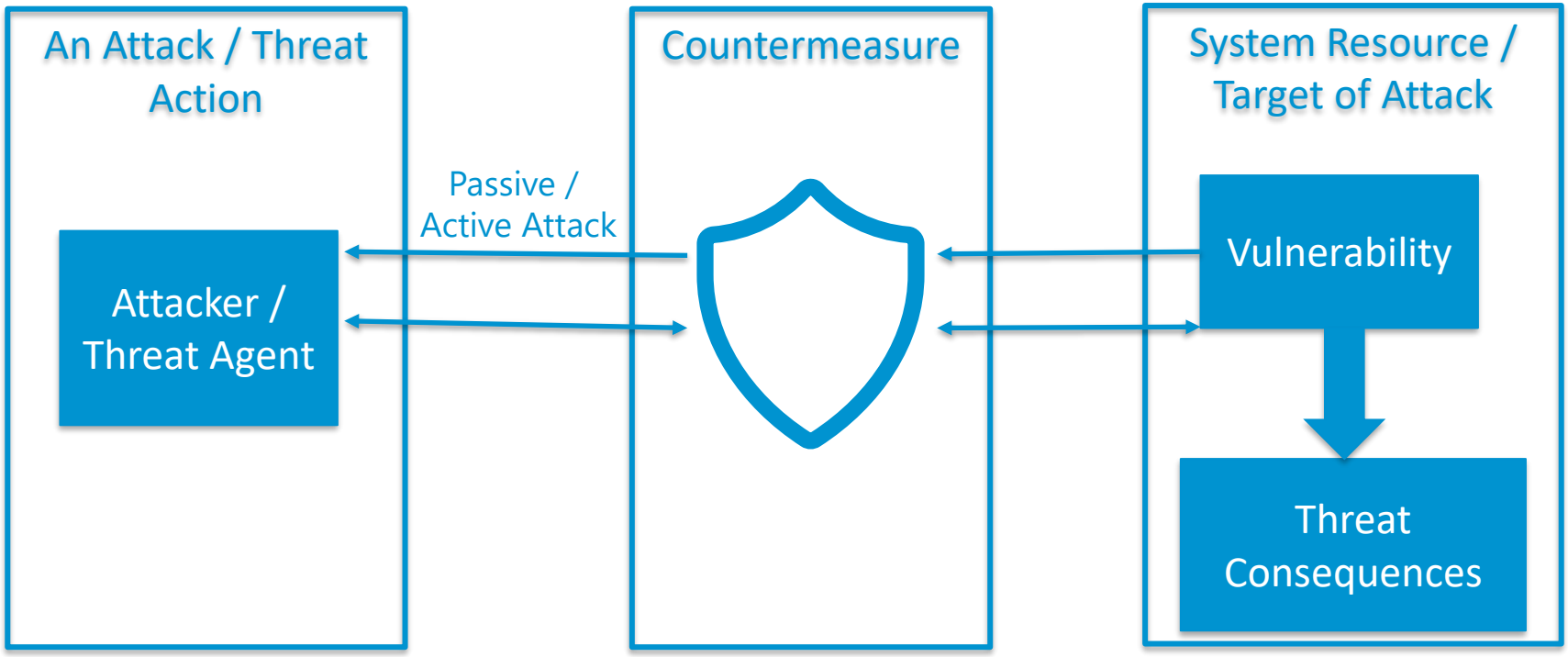
Controls & Mitigations

Controls *are safeguards or countermeasures that you put in place in order to avoid, detect, counteract, or minimize potential threats against your information, systems, or other assets.*

Mitigations *are controls that are put in place to reduce either the likelihood or the impact of a threat, while not necessarily completely preventing it.*

https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

Attack

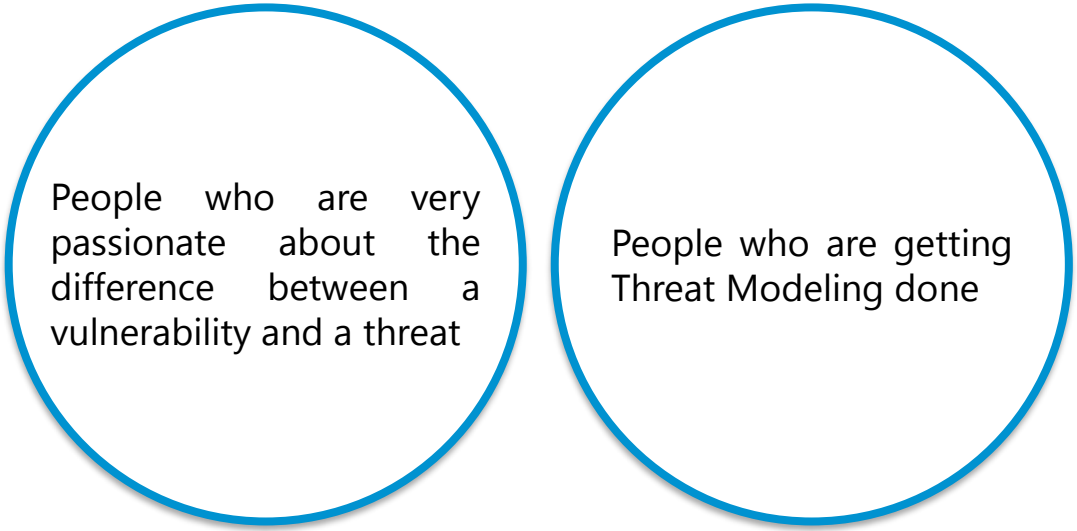


https://datatracker.ietf.org/doc/html/rfc2828

Example

Vulnerability	Passwords stored in plain text form in database
Threat	A DB-admin (= *threat actor*) copies all passwords from the database and leaks them online
Risk	Customer loss due to bad press following the leak

Vagueness


People who are very passionate about the difference between a vulnerability and a threat
People who are getting Threat Modeling done

Step 1

What are we building?

Modeling your system

All models are wrong, some models are useful.

-- George Box, FRS; quoted by Adam Shostack

Photo from https://adam.shostack.org

Context

To make an effective model of an application, you first need to understand its context

- Applications have connections with other systems
- There is a technical context and a business context
- Who are the users?
- What are they doing with the application?
- What are the requirements & assumptions?

Assumptions & Requirements

- What are the requirements of the product?
  - Business objectives
  - Legal / Compliance
  - Standards
  - Internal
  - …
- Under which assumptions do you operate?
  - How are things happening (“We use TLS everywhere”)
  - What is not happening (ADMIN_NO_EVIL)
  - How is the product used
  - …

Doomsday Scenarios

Define the 2-3 worst case scenarios for your product.

Photo by Clément Falize on Unsplash

Meet MedicalTechLab2000

- MedicalTechLab2000 (MTL2K) is a company that operates laboratories which evaluates tests for multiple viruses.
- Testee register with their personal information (name, phone number, e-mail address, social security number, address, birthday).
- A test kit is registered with a web application when the test is performed and picked up by a bike-courier.
- After the tests are processed, a download link for the report is sent via e-mail.

Doomsday Scenarios

What are the doomsday scenarios for MedicalTechLab2000?

Discuss it in groups and present your solutions.

Assets

Good start is to compile a list of *assets*:

- Things attacker want
- Things you want protected
- Mark the particularly valuable (PII, credentials, …)

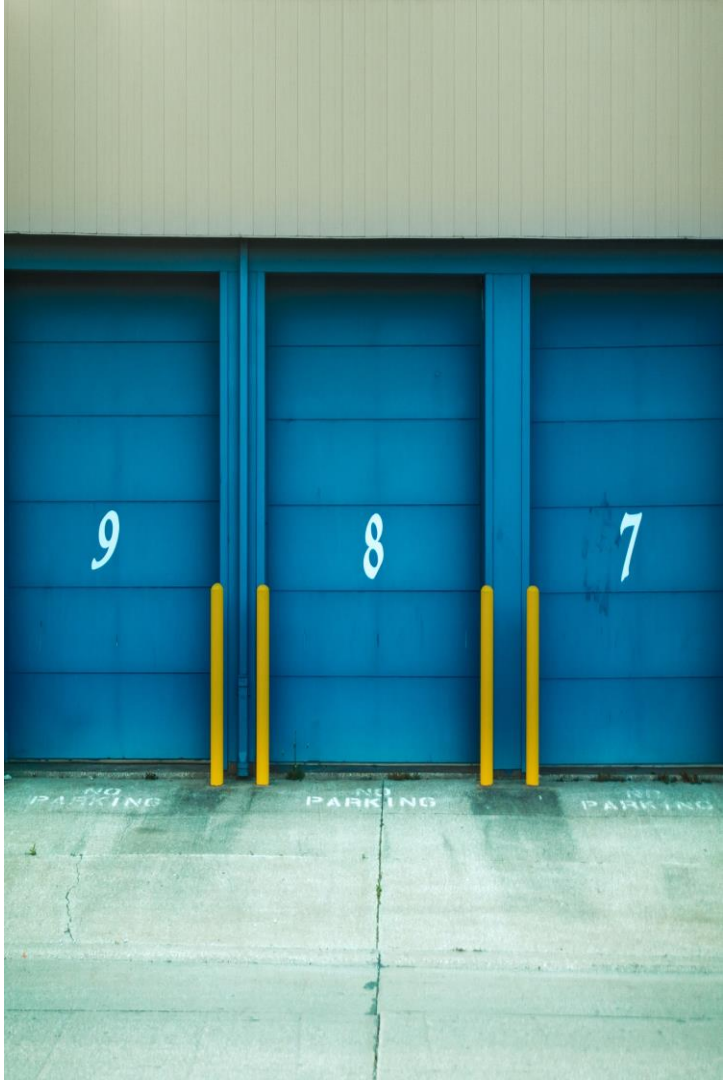


Photo by Steve Johnson on Unsplash

Dataflow Diagram

- Diagram of the system
- Shows where data is
  - Stored
  - Used
  - Transferred

Diagram Levels

- Context diagram (L0)
  - Whole system
- High level diagram (L1)
  - Either whole system or single features
- Detailed diagrams (L2)
  - Low Level
  - Detailed components of single feature

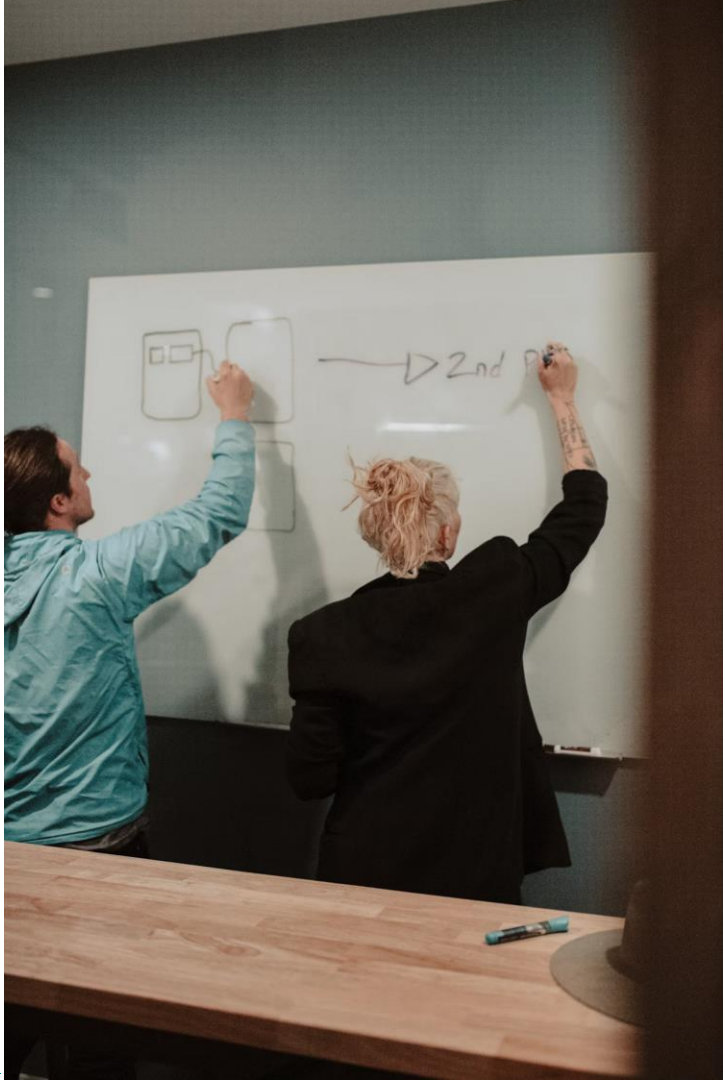
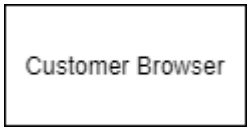


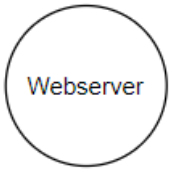
Photo by Sandra Seitamaa on Unsplash

Diagram Elements



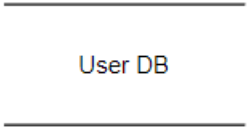
External entity

Trust boundary



Process

Data flow



Datastore

Trust Boundary

A location in the diagram, where data changes its level of trust or ownership.

Defined by a *policy* (what is allowed) and *controls* (how is the policy enforced).

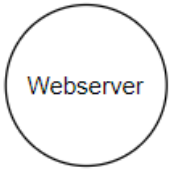
Can be strong or weak.

- Calling a 3rd party API
- Reading user input
- Webserver reading from a database

Process

A *Process* **uses** data.

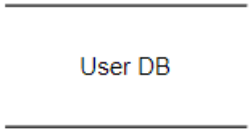
At least one incoming and one outgoing dataflow



Data Store

A *Data Store* **stores** data.

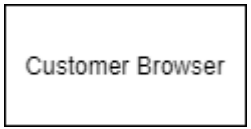
At least one incoming and one outgoing dataflow



External Entities

Not controlled by the application or system.

Must be connected to some process



Diagram

- Living objects; update when system changes
- Label everything (number, ID or name)
  - Can be referenced in other parts of the model (threat list)
- Better to make multiple small than one huge diagram
- Auxiliary diagrams can help you (business flows, sequence diagrams, architecture, …)

Diagram

- Use a whiteboard for collaborative work
  - Actual whiteboard
  - Digital whiteboard e.g., https://excalidraw.com
- Make a „pretty version“ in a diagram tool of your choice after you are finished
  - e.g., https://diagrams.net (a.k.a. *draw.io*)

Diagram Example
Simple Webshop: Level 0

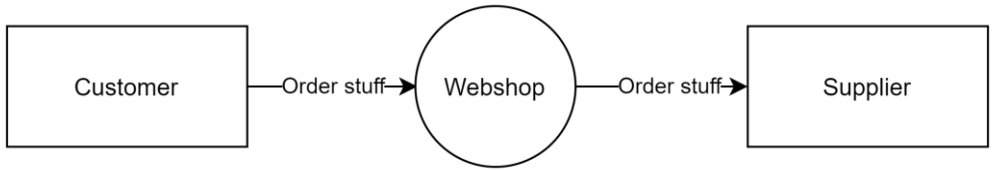


Diagram Example

Simple Webshop: Level 1

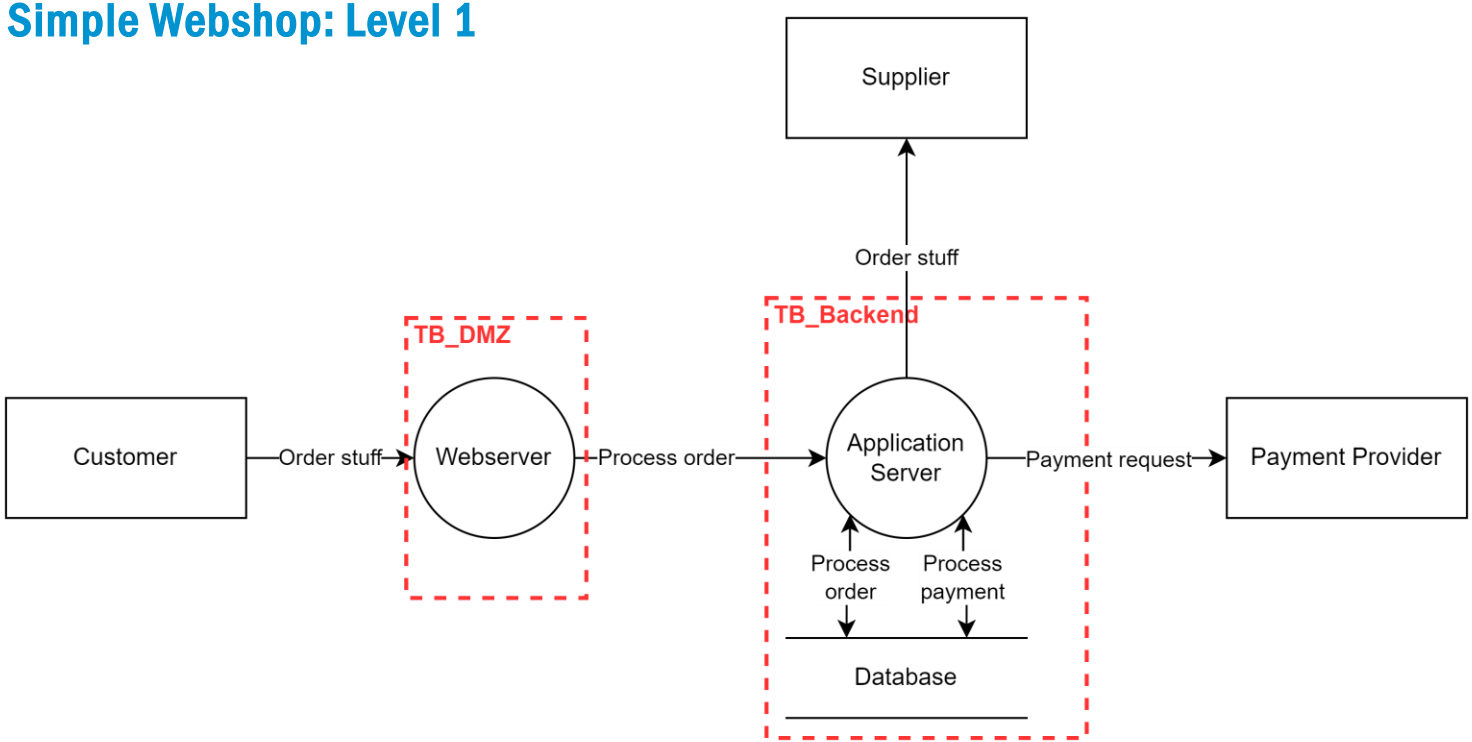
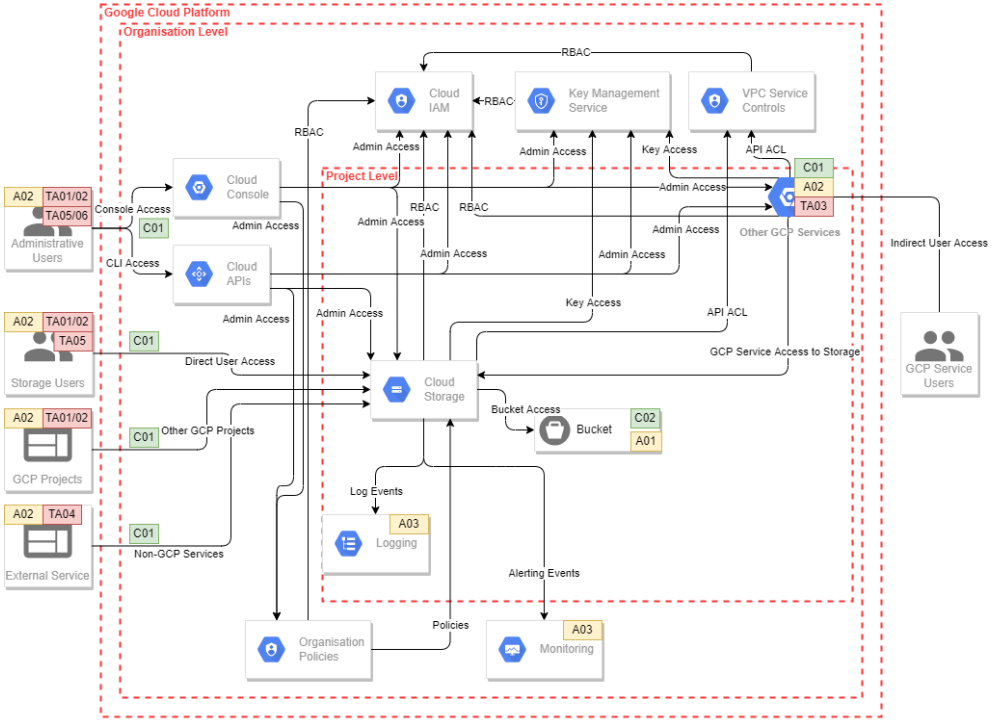


Diagram Example



Assets	
ID	Description
A01	Bucket object data
A02	Authentication tokens
A03	Log Data

Threat Actors	
ID	Description
TA01	Internal attacker
TA02	Internal malicious user
TA03	Compromised internal service
TA04	Compromised external service
TA05	External attacker over the Internet
TA06	Google Engineers

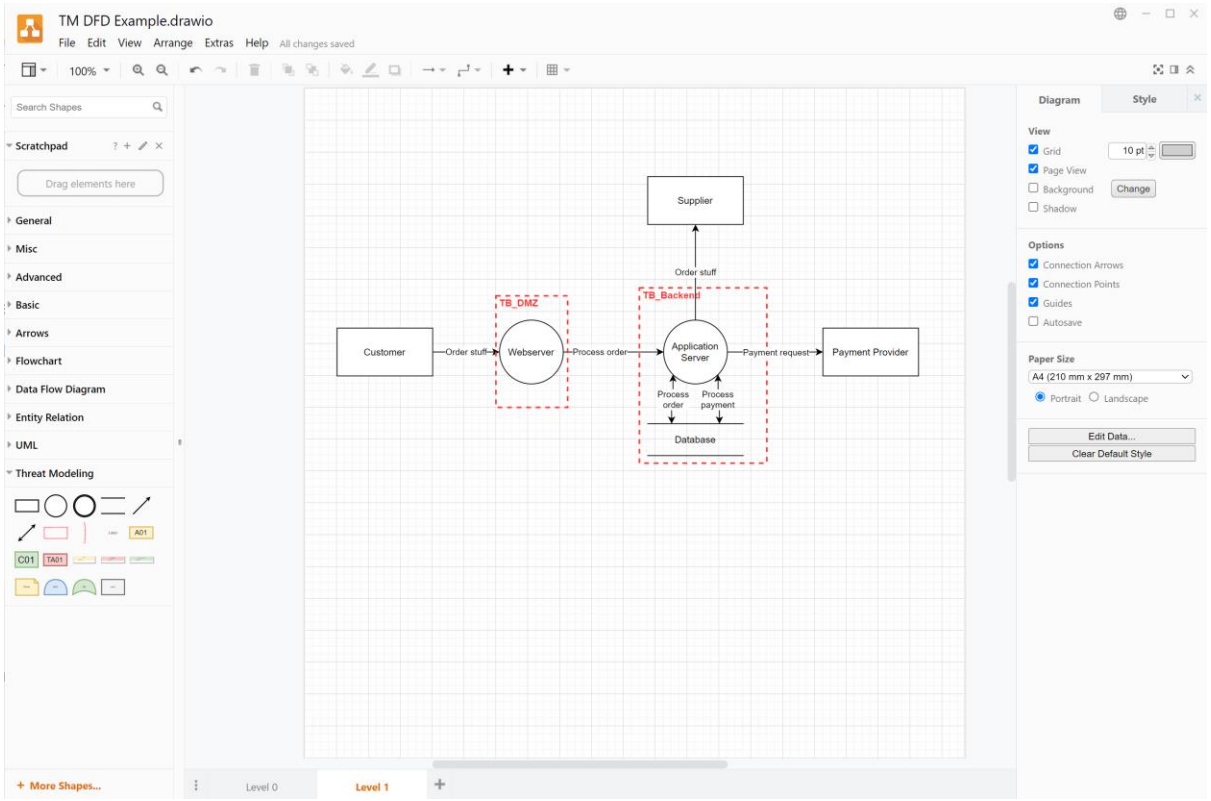
Default Base Security Controls	
ID	Description
C01	HTTPS transport layer encryption
C02	Data-at-rest encryption

Source: https://research.nccgroup.com/2023/01/31/threat-modelling-cloud-platform-services-by-example-google-cloud-storage/

Tool

Diagrams.net a.k.a. draw.io

https://app.diagrams.net/



Create model

Create a model (level 0 and level 1) for MedicalTechLab2000.

Draw a DFD in groups and present your solutions.

Step 2

What can go wrong?

Identify Threats

- Finding threats by analyzing the diagram
- Threats can be
  - On a component (process, entity, datastore)
  - Where a dataflow crosses a trust boundary
- Use STRIDE to find threats

Photo by Jason Leung on Unsplash

STRIDE

- Model to help you identify threats
- Developed at Microsoft in 1999
- Mnemonic of its 6 threat types

STRIDE

Spoofing

Tampering with data

Repudiation

Information disclosure

Denial of service

Elevation of privileges

STRIDE
Spoofing

- Performing an action while pretending to be someone else
- Defeats *Authentication*
  - Also reduces *Accountability*

- Missing or weak authentication
- Account takeover
- Bad session management

STRIDE
Tampering

- Altering data that you are not allowed to change
- Defeats *Integrity*

- Injection attacks
- Cross-Site-Scripting (XSS)
- Cross-Site-Request-Forgery (CSRF)

STRIDE
Repudiation

- Successfully denying having performed an action
- Defeats *Non-Repudiation*

- Insufficient logging
- Logs can be tampered with
- Missing access control

For a privacy relevant part of your application, this might actually be desired and not a threat!

STRIDE
Information Disclosure

- Retrieving data not intended for one
- Defeats *Confidentiality*

- Missing access control
- Missing encryption
- Sensitive data in logs or backups

STRIDE
Denial-of-Service

- Prevent access to a service by legitimate users
- Defeats *Availability*

- Missing DDoS protection
- Bad account lockout policy
- Ransomware attack

STRIDE
Elevation of Privileges

- I can get access to a system with higher privileges than intended
- Defeats *Authorization*

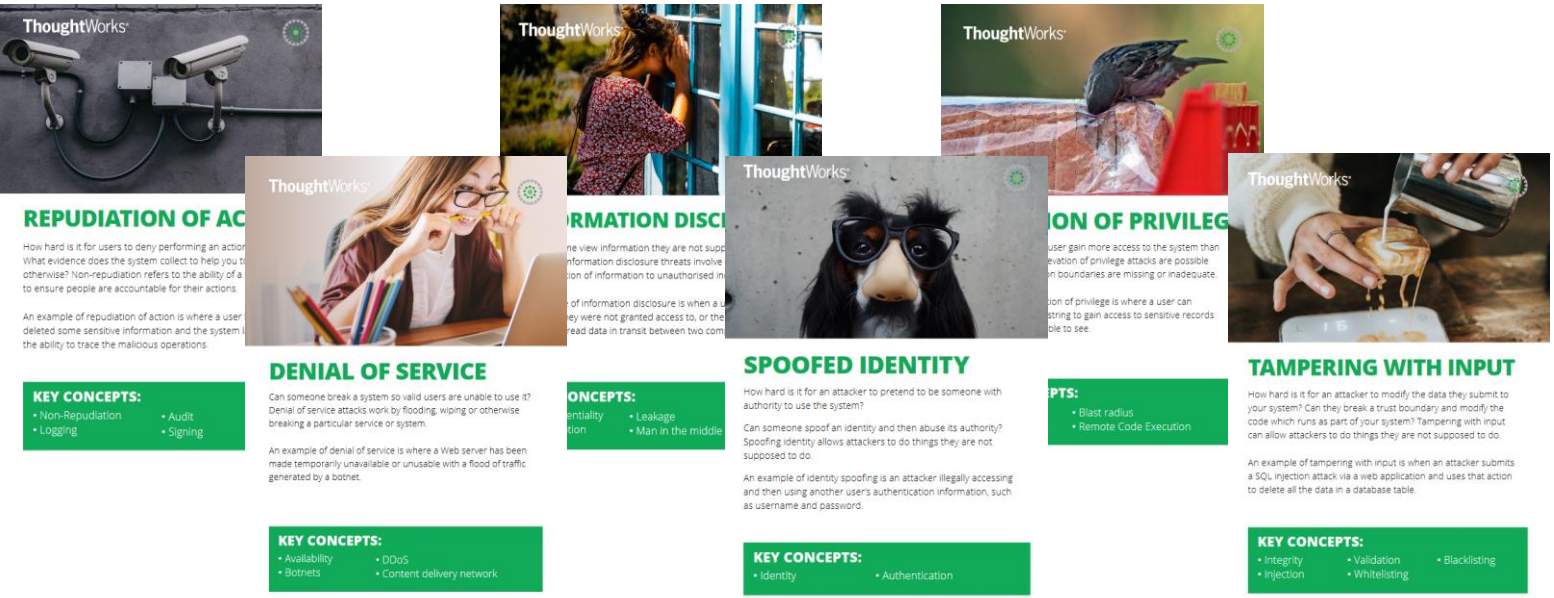
- RCE on webserver
- Server-Side-Request-Forgery (SSRF) to internal network
- Broken access control

STRIDE

Spoofing	Authentication
Tampering with data	Integrity
Repudiation	Non-Repudiation
Information disclosure	Confidentiality
Denial of service	Availability
Elevation of privileges	Authorization

STRIDE cards

Double sided printable cards to help you with the STRIDE elements



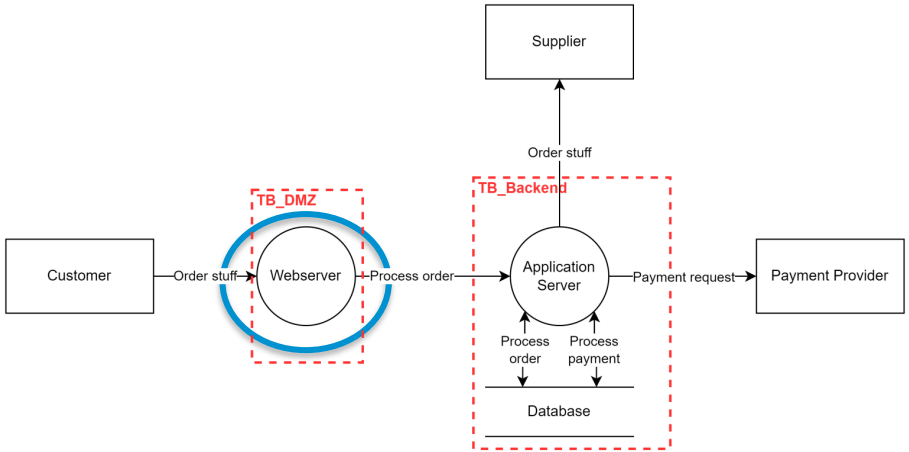
https://thoughtworksinc.github.io/sensible-security-conversations/materials/Sensible_Agile_Threat_Modelling_Cards.pdf

Finding Threats

Look for threats at the elements of your diagram

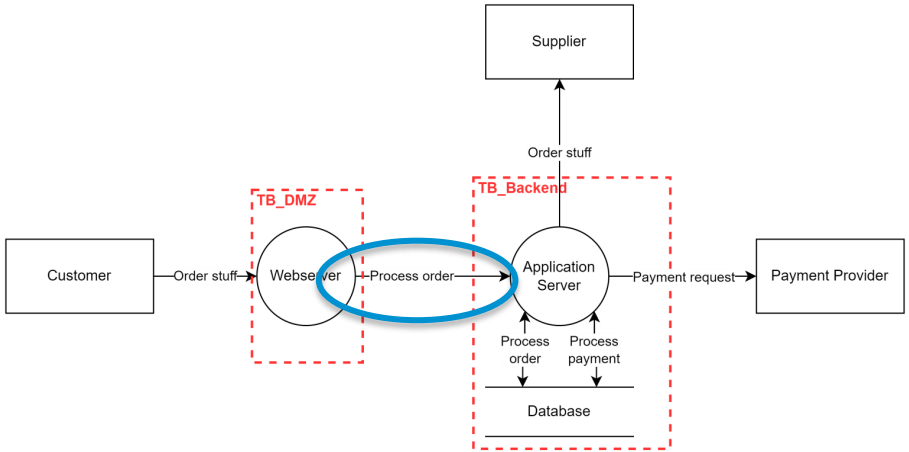
1. Data flows crossing a trust boundary
2. Other data flows
3. Processes
4. Data storage
5. (External entities)

Sample Threats
Spoofing



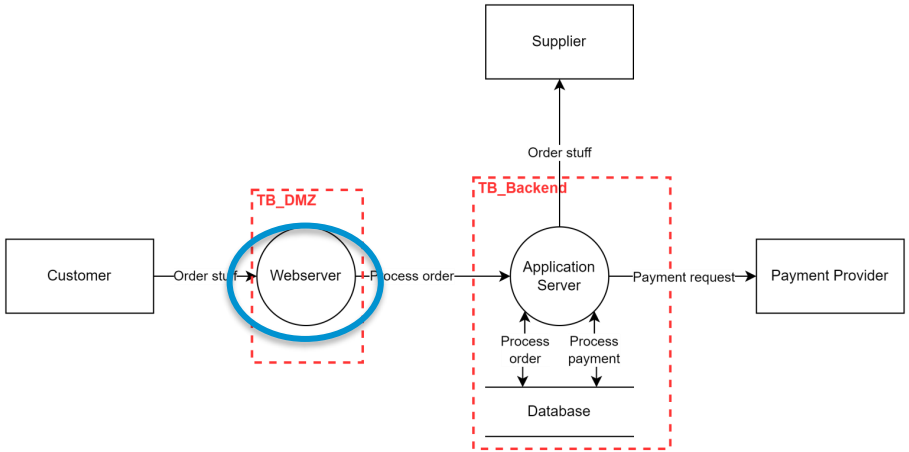
Title	STRIDE	Description
Account takeover	S	Password-reset functionality allows an account takeover via security question

Sample Threats
Tampering



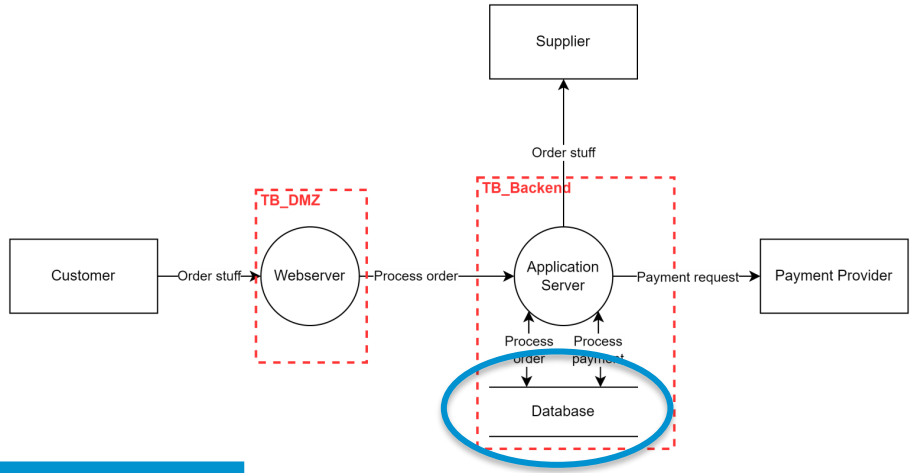
Title	STRIDE	Description
MitM	TI	Data gets altered on an unencrypted network connection

Sample Threats
Repudiation



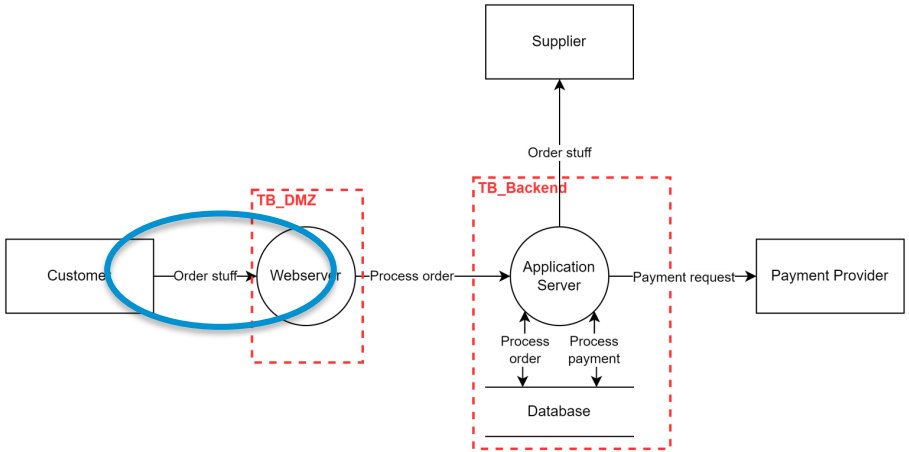
Title	STRIDE	Description
Insufficient Logging	R	Frontend logging is very limited, attacks can go unnoticed

Sample Threats
Information Disclosure



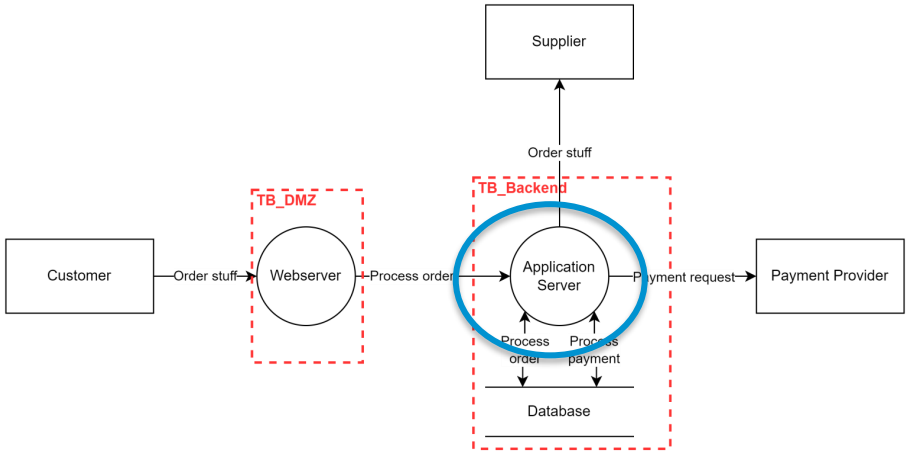
Title	STRIDE	Description
Bad DB credentials	I	Simple DB password is known throughout the company

Sample Threats
Denial-of-Service



Title	STRIDE	Description
Lockout	D	Brute-forcing the login causes a user-lockout

Sample Threats
Elevation of Privileges



Title	STRIDE	Description
Broken AuthZ	E	Missing function level access control allows usage of administrative functions

STRIDE-per-Element

Useful method for identifying the most common threats.

	S	T	R	I	D	E
Data Flow						
Process						
Data Store						
External Entity						

Threat actors

- Define attackers and their capabilities & motivations
  - Internal attacker (employee)
  - Passive network observer
  - …
- Must not lead to ignoring threats not falling into one of those buckets

Threat actors

Map threat actors to the threats you have identified.

	TA1 (Network operator)	TA2 (Normal user)	TA3 (Evil admin)	TA4 (Employee)
T1 (MitM)				
T2 (Brute force login)				
T3 (DB credentials)				
T4 (Abuse payment service)				

Threat Types

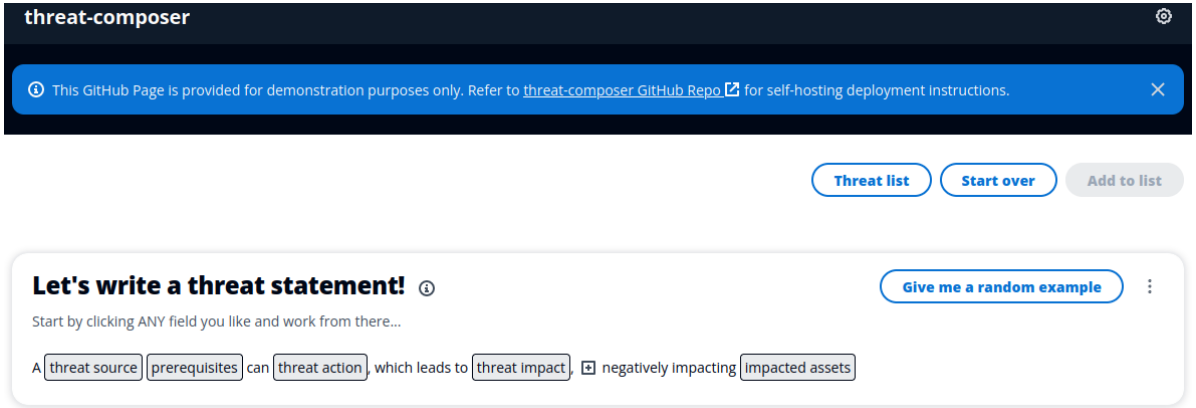
- Generic threats vs. specific threats
  - “MitM attack on unencrypted network connection”
  - “Excessive customer data sent to payment provider”
- Combination of multiple threats/vulnerabilities may yield larger, systemic threats
- Accidental vs. Tradeoff vs. Inherent threats
  - Adam Shostack: Inherent Threats – Clarifying a property of threats

Threat Documentation

- Focus on most prevalent scenarios/areas.
- Be specific: not “data gets dumped”, no catch-all (“system compromised”)
  - Think about mitigations
  - (Think about risk rating)
- Don’t forget the threat actors
  - Result: table of threat actors
- Duplicate threats with different threat actors are possible

Threat Documentation

threat-composer



Let's write a threat statement! Give me a random example

Start by clicking ANY field you like and work from there...

An external threat actor with a network path to the database can read other user's documents, which leads to a compliance violation, resulting in reduced confidentiality of invoices

Threat Documentation

ID	Title	STRIDE	Threat Actor	Location	Description
T1	Account takeover	S	Internet user	P1	Password-reset functionality allows an account takeover via security question
T2	MitM	TI	Internal network member	DF1	Data gets altered on an unencrypted network connection
T3	Bad DB credentials	I	Employee	DS1	Simple DB password is known throughout the company

Find Threats

Find relevant threats for your model of MedicalTechLab2000.

Discuss it in groups and present your solutions.

ID	Title	Location	STRIDE	Threat Actor	Description
T52	Broken AuthZ	P1	E	Registered User	Missing function level access control allows usage of administrative functions

Step 3

What can we do about it?

Risk

- Every threat adds a *Risk* to the system
- *Risk* is a quantification of a threat which helps you prioritize them
- Risk = Likelihood * Impact
  - Technical Impact
  - Business Impact

Mitigations / Controls

Ways to address your threats or at least parts of them.

Storage

Every device and field gateway has some form of storage (temporary for queuing the data, operating system (OS) image storage).

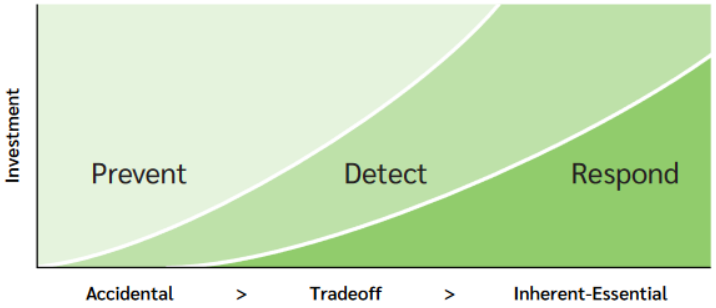
Component	Threat	Mitigation	Risk	Implementation
Device storage	TRID	Storage encryption, signing the logs	Reading data from the storage (PII data), tampering with telemetry data. Tampering with queued or cached command control data. Tampering with configuration or firmware update packages while cached or queued locally can lead to OS and/or system components being compromised	Encryption, message authentication code (MAC), or digital signature. Where possible, strong access control through resource access control lists (ACLs) or permissions.
Device OS image	TRID		Tampering with OS /replacing the OS components	Read-only OS partition, signed OS image, Encryption

Types of Controls

- Preventive
  - Security door
  - TLS for network connection
- Detective
  - Motion sensors
  - Monitoring / Incident detection
- Corrective
  - Fire extinguisher
  - Isolate threat actor, patch vulnerability

Mitigations / Controls

- Try to remove a threat
- More likely: Reduce its risk
  - Reduce either the likelihood or the impact of a threat
- Different control types, depending on the threat type



https://shostack.org/files/papers/Inherent-Threats-Whitepaper-Shostack.pdf

Threat Documentation

Mitigations

ID	Title	STRIDE	Threat Actor	Loc.	Description	Existing Mitigations	Suggested Mitigations
T1	Account takeover	S	Internet user	P1	Password-reset functionality allows an account takeover via security question	None	Do not use security questions Rate Limiting per IP and user
T2	MitM	TI	Internal network member	DF1	Data gets altered on an unencrypted network connection	Switch prevents ARP spoofing	Use TLS
T3	Bad DB credentials	I	Employee	DS1	Simple DB password is known throughout the company	DB logins are logged	Use high entropy password, rotate regularly

Security Requirements

Derive requirements from your mitigations:

- Password reset must only be done via a reset link sent to a known e-mail address
- Every IP address must no request more than 1 password reset every 15 minutes
- The network connections between frontend and backend must use TLS
- All passwords for internal services must be at least 12 characters long
- All service account passwords for internal services must be at least 25 characters long
- All shared passwords must be rotated every 3 months

Process

Risk management

1. Identify threats
2. Calculate risks for all threats
3. **Treat risks**
4. Implement mitigations
5. Calculate residual risk
6. Go back to 3.

What happens with a risk?

You can do either of the following things with a risk:

- Avoid
- Reduce/Address
- Transfer
- Accept
- ~~Ignore~~

Photo by Michael Shannon on Unsplash

What to fix

- Prioritize what to fix first
  - Severe risks should usually be fixed first as it does barely improve the overall risk profile fixing low rated issues.
  - Not all risks are worth fixing as the investment to fix them would be too high.
  - *Cost-to-Fix* is also important

Prioritization

You will need to find a way to rank your threats, to find out which ones to deal with first

1. Voting
2. Quantify risk (Critical – High – Medium – Low – Info)
    1. Ad-Hoc
    2. Sophisticated

Risk calculation

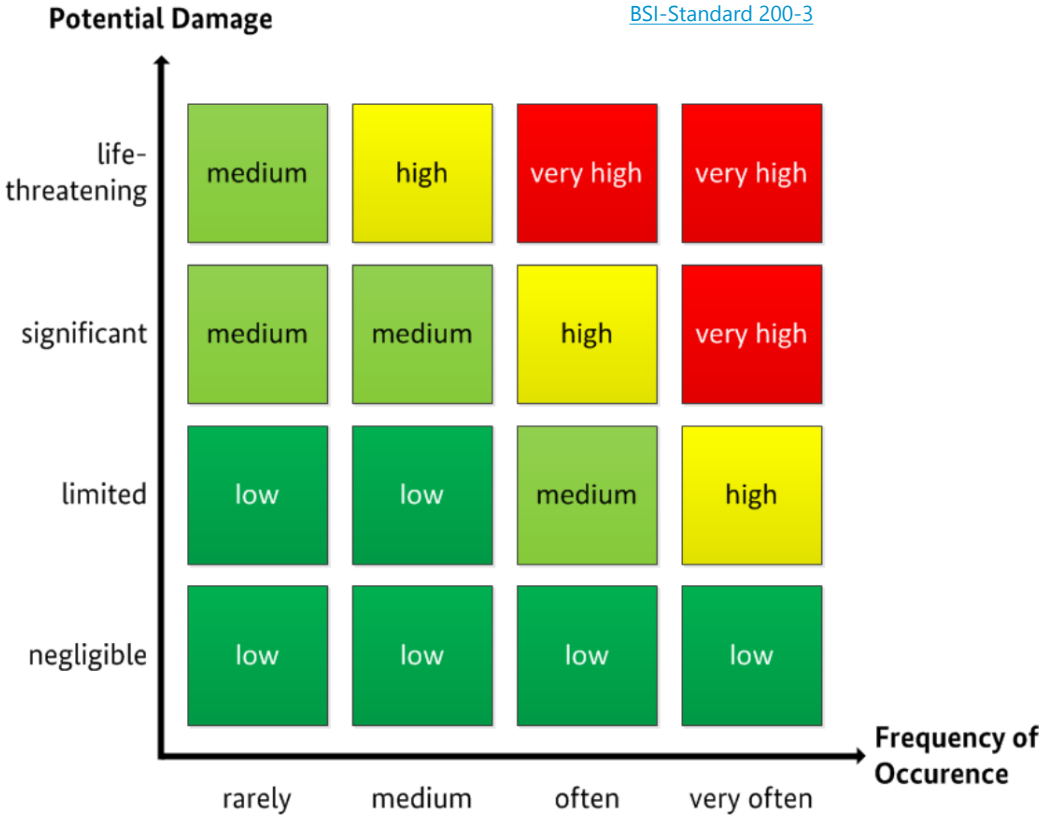
3x3 Risk Matrix

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Risk calculation

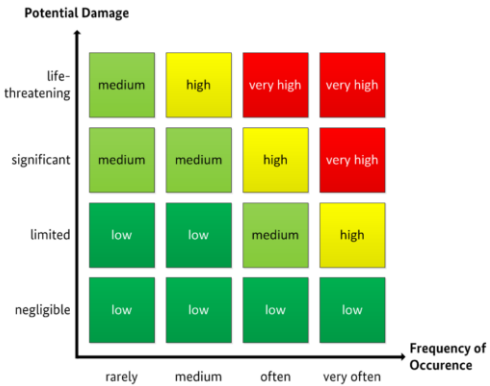
4x4 Risk Matrix (e.g. BSI-Standard 200-3)

BSI-Standard 200-3



Risk calculation

BSI-Standard 200-3



BSI-Standard 200-3

Frequency	Description
Rarely	According to present knowledge, the event could occur every 5 years at the most
Medium	The event occurs once every 5 years to once a year
Often	The event occurs once a year to once a month
Very Often	The event occurs several times a month

Damage	Description
Negligible	The effects of damage are low and can be neglected
Limited	The effects of the damage are limited and manageable
Significant	The effects of damage can be considerable
Life Threatening	The effects of the damage can reach a catastrophic level that threatens the existence of the organisation

Identify Top Threats

Identify the most relevant threats (~5) for your model of MedicalTechLab2000.

Discuss it in groups and present your solutions.

Risk calculation

OWASP Risk Rating Methodology

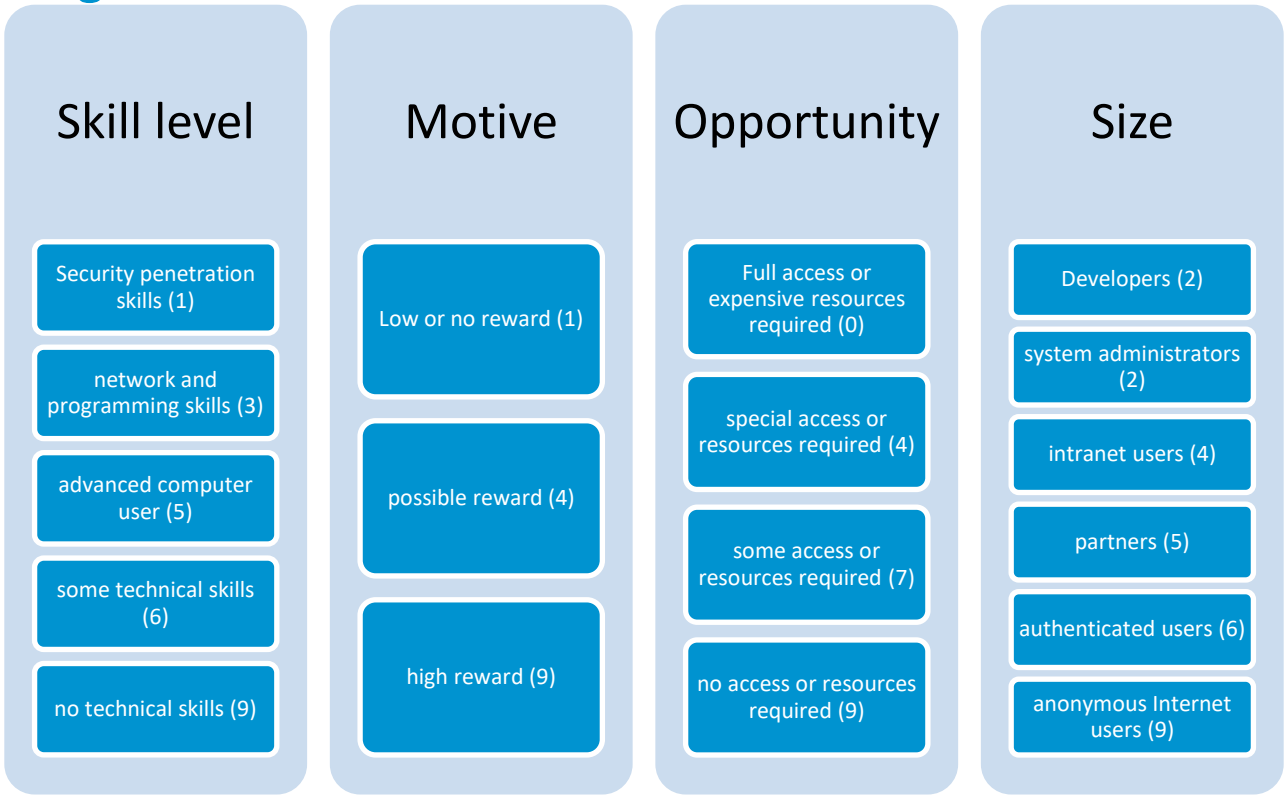
Slightly more complex way of quantifying a risk.

The methodology consists of 6 steps:

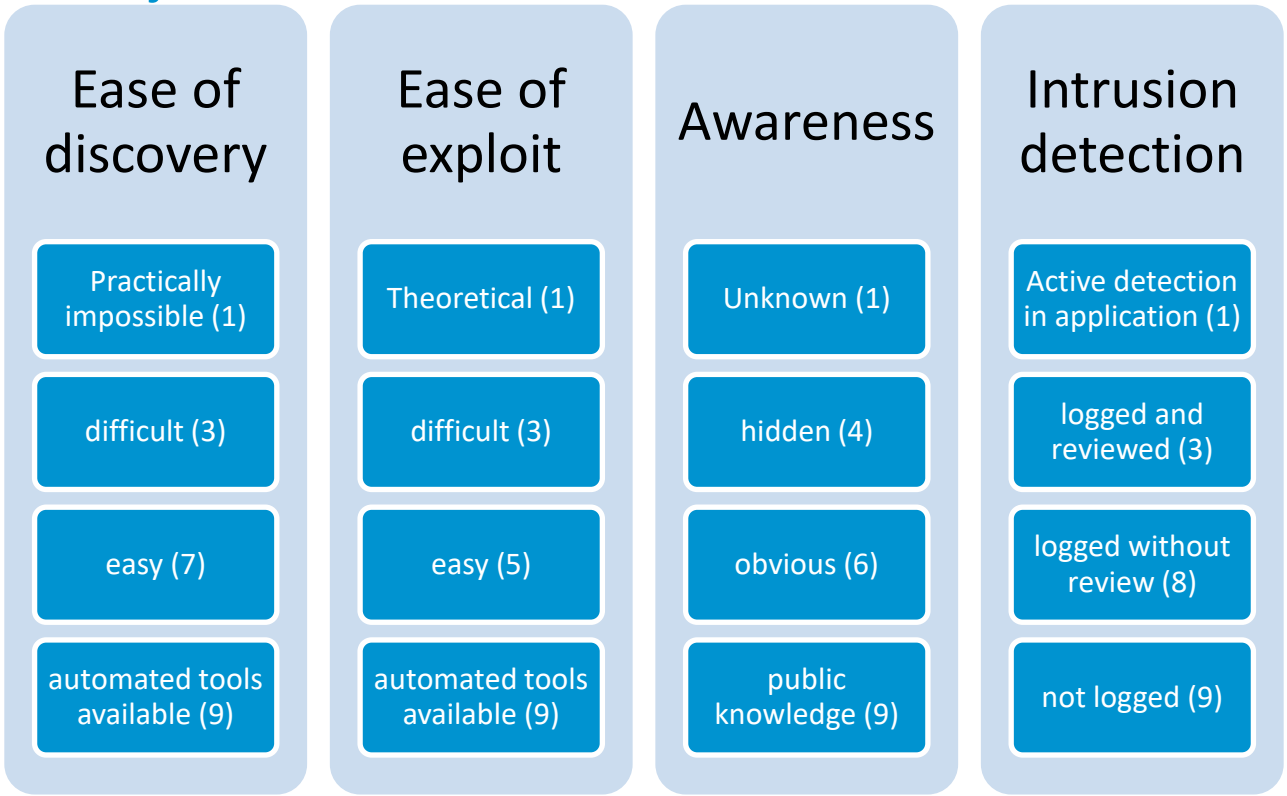
1. Identifying a Risk
2. Factors for Estimating Likelihood
3. Factors for Estimating Impact
4. Determining Severity of the Risk
5. Deciding What to Fix
6. Customizing Your Risk Rating Model

Estimating Likelihood

Threat Agent Factors

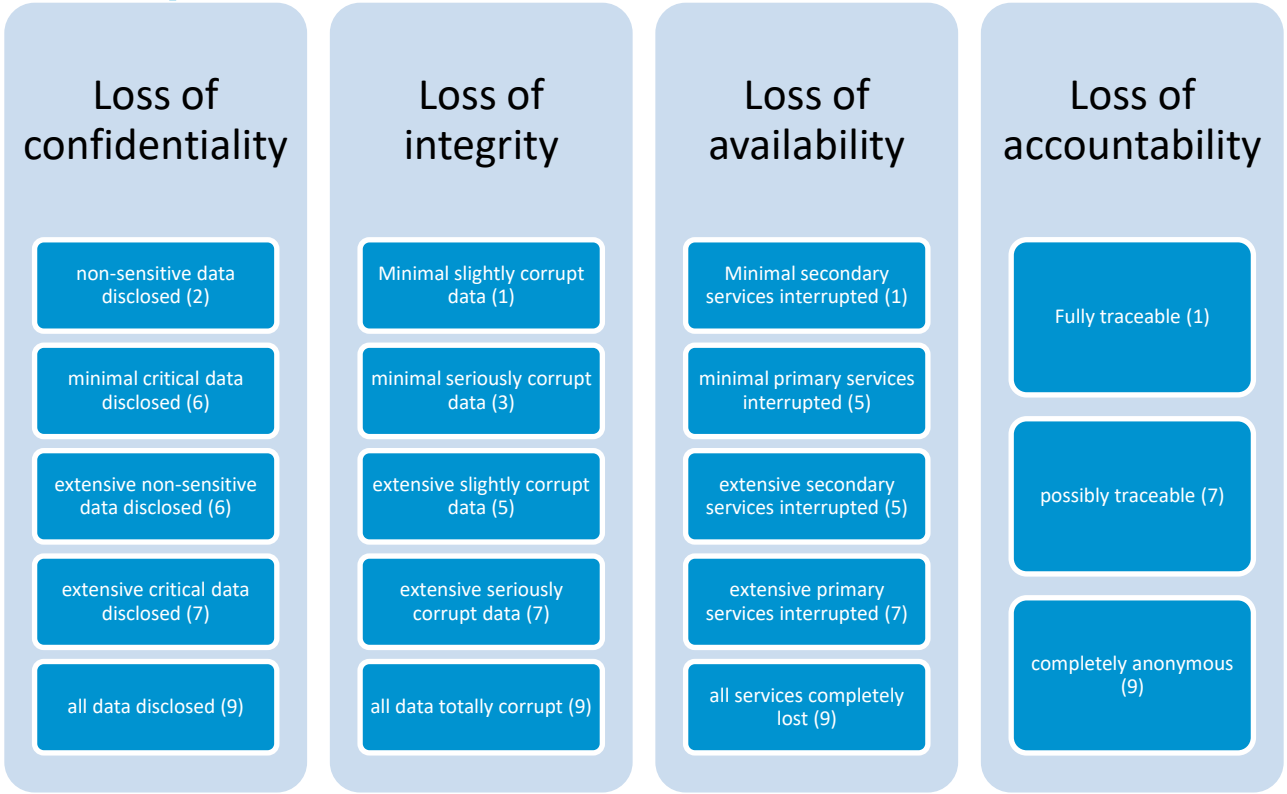


Estimating Likelihood
Vulnerability Factors

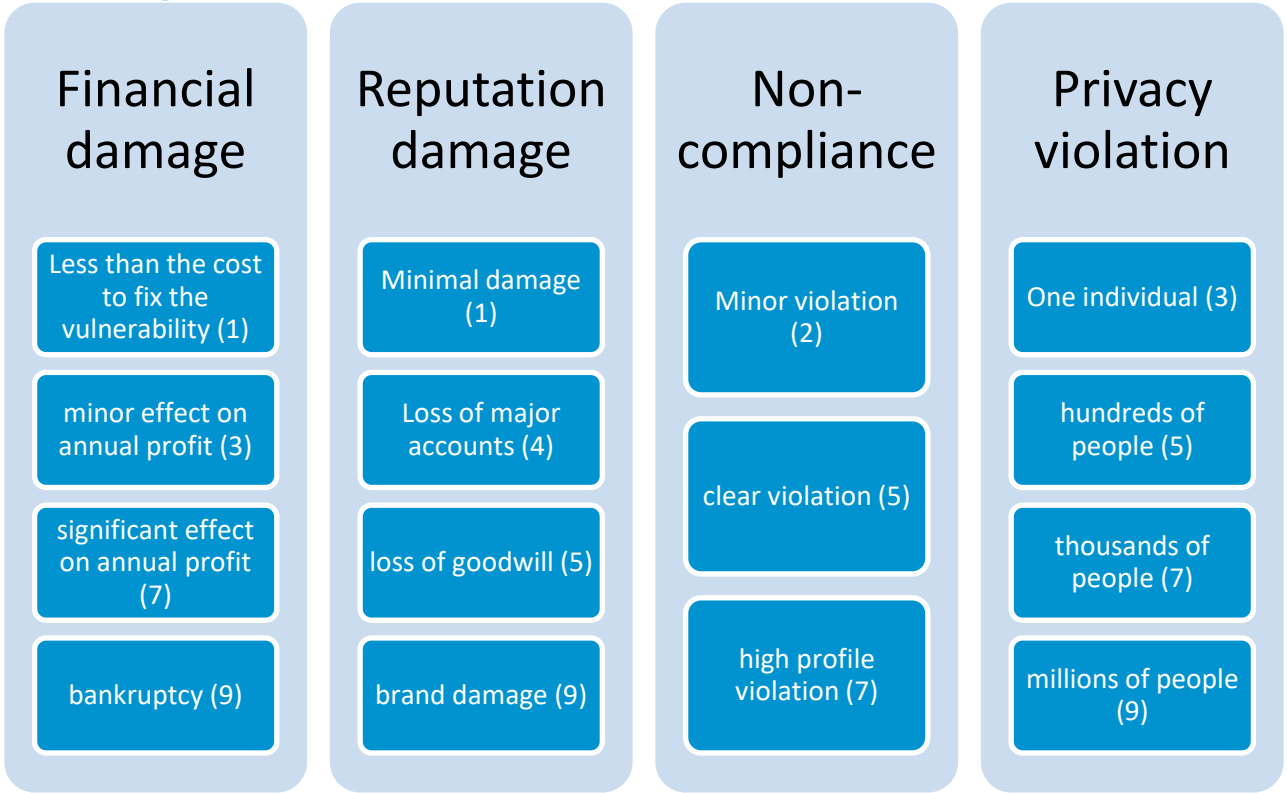


Estimating Impact

Technical Impact Factors



Estimating Impact
Business Impact Factors



Determining Severity

- Calculate mean value of all ratings (likelihood & impact)
  - 0–3 LOW
  - 3–6 MEDIUM
  - 6–9 HIGH
- Derive overall severity

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Toolbox

Likelihood								
Threat agent factors					Vulnerability factors			
Skill level	Motive	Opportunity	Size		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
4 - Advanced computer user	1 - Low or no reward	4 - Special access or resources required	5 - Partners		3 - Difficult	5 - Easy	4 - Hidden	3 - Logged and reviewed
				3,625	MEDIUM			

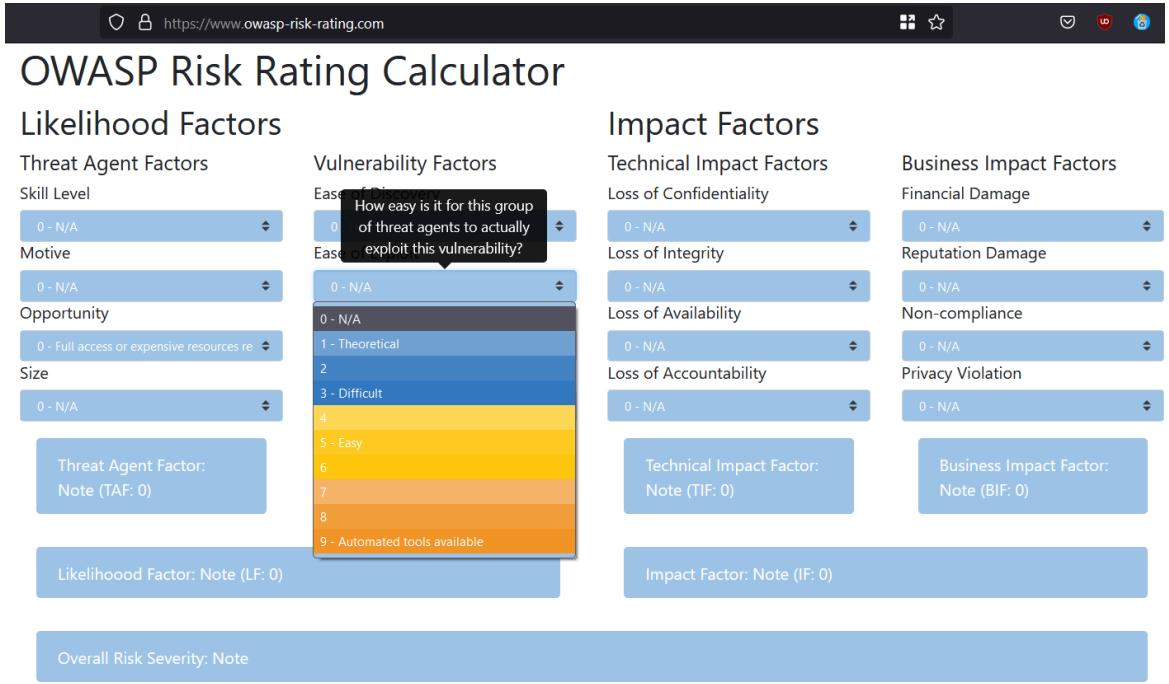
2 - Developers, system a
3 -
4 - Intranet users
5 - Partners
6 - Authenticated users
7 -
8 -
9 - Anonymous Internet

Technical Impact					Business Impact			
Loss of confidentiality	Loss of integrity	Loss of avail			Financial damage	Reputation damage	Non-compliance	Privacy violation
2 - Minimal non-sensitive data disclosed	0 -	0 -	anonymous		1 - Less than the cost to fix the vulnerability	1 - Minimal damage	0 -	5 - Hundreds of people
Overall technical impact:		2,750	LOW		Overall business impact:		1,750	LOW
Overall impact:				2,250	LOW			

Overall Risk Severity = Likelihood x Impact				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Toolbox



https://www.owasp-risk-rating.com/

Customization

Possible customizations of the risk rating model

- Adding factors
  - Based on the organization or application, different factors may be of interest
- Customizing options
  - The description or the weighting of different options can be adapted
- Weighting factors
  - The weight of different factors can be adapted to the specific business situation

Threat Documentation

Risk

ID	Title	STRIDE	Threat Actor	Loc.	Description	Existing Mitigations	Suggested Mitigations	Risk
T1	Account takeover	S	Internet user	P1	Password-reset functionality allows an account takeover via security questions	None	Do not use security questions Rate Limiting per IP and user	HIGH (I:M L:H)
T2	MitM	T I	Int. network member	DF1	Data gets altered on an unencrypted network connection	Switch prevents ARP spoofing	Use TLS	LOW (I:M L:L)
T3	Bad DB credentials	I	Employee	DS1	Simple DB password is known throughout the company	DB logins are logged	Use high entropy password, rotate regularly	MEDIUM (SL:5/M:4/O:7/S:4/ED:7/EE:9/A:6/ID:3/LC:9/LI:0/LAV:0/LA

Quantify Top Threats

Use the OWASP methodology to calculate the risk level of your top threats.

Thank You

Congratulations on completing your first steps into the world of Threat Modeling!

Feedback:
aboll@sba-research.org

Photo by George Pagan III on Unsplash

Andreas Boll

SBA Research

Floragasse 7, 1040 Wien

aboll@sba-research.org