Threat modeling workshop

Practice session hangouts

Sirris - SecDes

Contents

- Part 1 – theory – WHY? And WHAT is IT?
- Part 2 – Practice - CONTEXT
- Part 3 – Practice - COMPONENT

Part 1: why to do it??

Design and secure design

- Specify requirements
- Implement features
- Build software people will use

- Specify SECURE requirements
- Implement SECURITY features
- Build software people will use

AND anticipate when something goes wrong

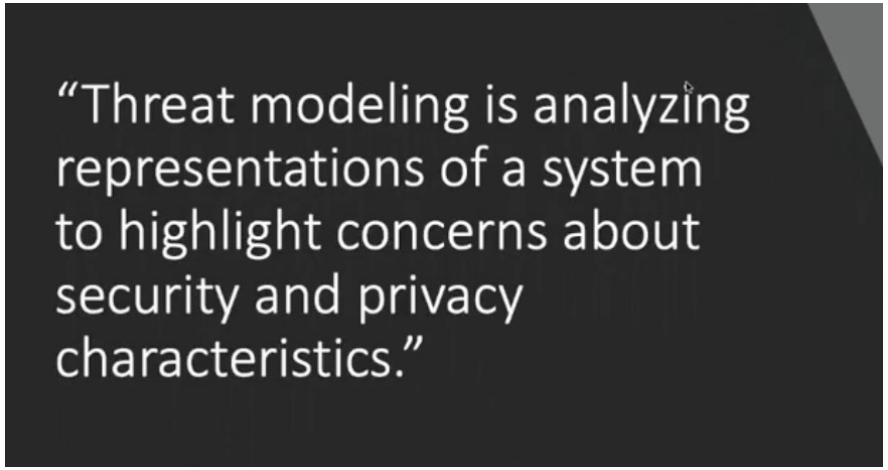
Examples

- How to secure data on prem? in the cloud?
- How to secure sensors in customer premises?

Secure design

- Thread modeling helps us to focus on these questions and answers

Secure design

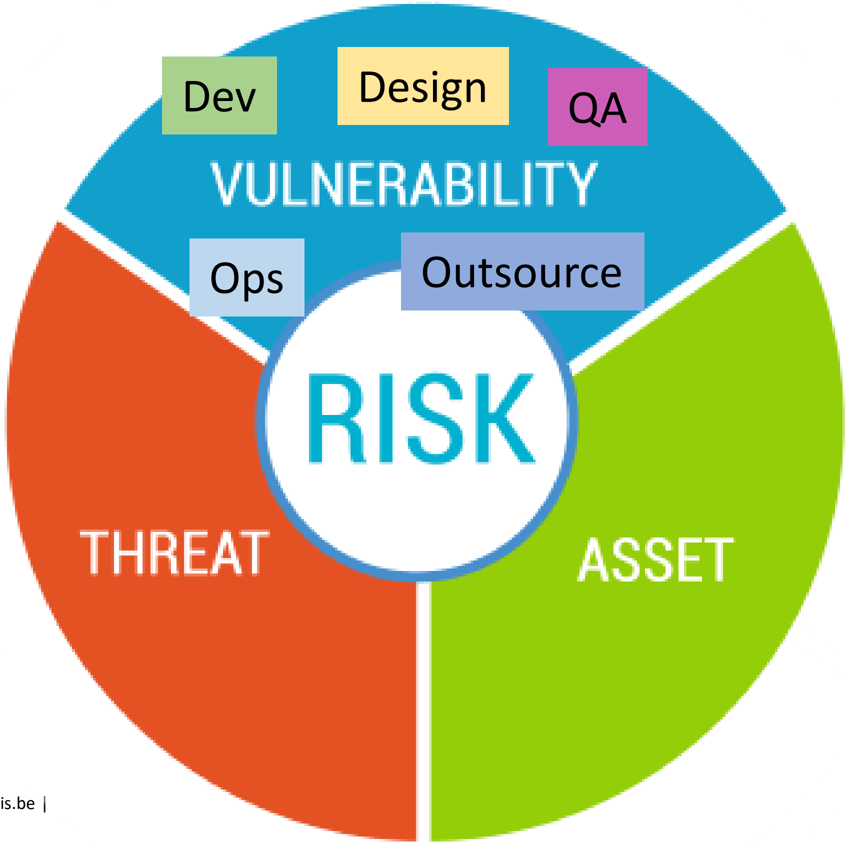


“Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.”

Ed Moyle (2017):
“Very few organizations will have the time or resources to ***threat model*** *their entire ecosystem.*
Assuming you do not have that luxury, you still can realize quite a bit of ***value***
just by adopting the mindset of looking for blind spots and questioning assumptions.” *
https://www.ecommercetimes.com/story/Invisible-Technologies-What-You- Cant-See-Can-Hurt-You-84852.html

How to be secure enough?

- Threat – potential to harm
- Vulnerability – weakness that can be used to harm
- Attack, vector, surface – threat realization scenario
- Likelibility – Chance for threat to happen
- Asset – what can be damaged
- Risk – how much you loose when this happens

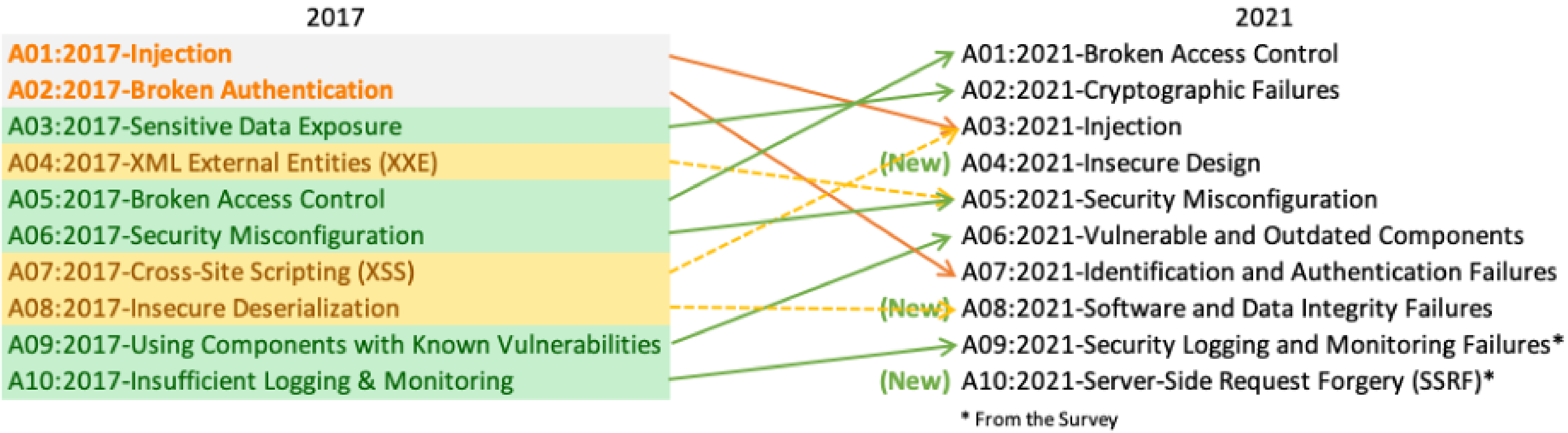


© sirris | www.sirris.be | info@sirris.be |

Know what and how protect!

How often do things go wrong?

- https://owasp.org/www-project-top-ten/ (TBD in 2025)



2017

A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*

* From the Survey

What do you already do for security of your software?

- ….

…if you do not do threat modeling – you miss a lot!

Locked?

Locked?

Think ahead

What if?

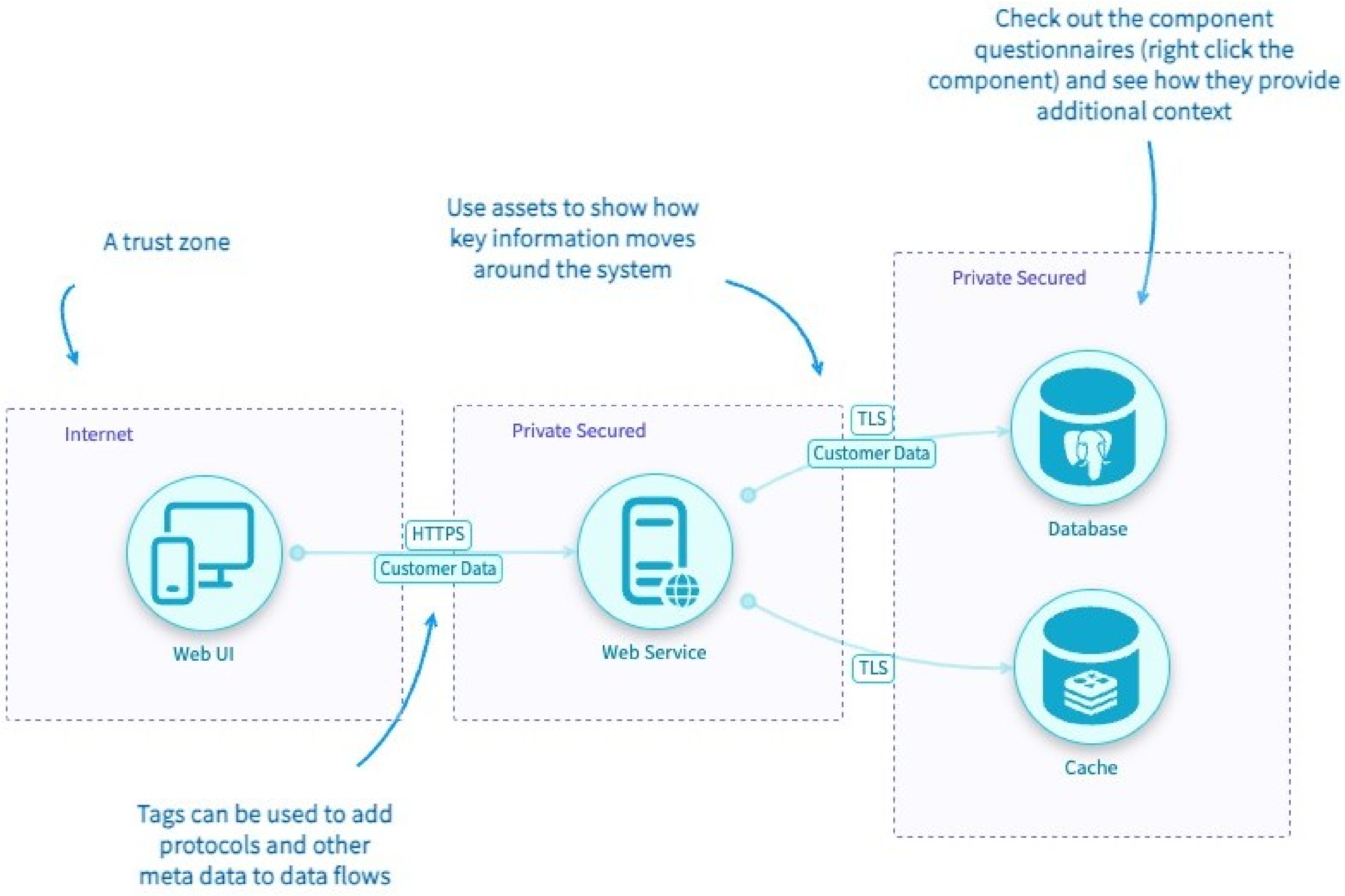
Weight the risks

Act accordingly

Threat modeling

- Process of understanding your system and potential threats against your system or Critical Security Thinking

> “Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics.”



Check out the component questionnaires (right click the component) and see how they provide additional context

A trust zone

Use assets to show how key information moves around the system

Tags can be used to add protocols and other meta data to data flows

Part 2: Context

Understand your system

DICE model

- Understand the system – and what stakeholders expect from it.
- Apply known successful attacks to points on a system where attackers can reach
- Rate the risk for each attack scenario
- Identify appropriate defenses or mitigations

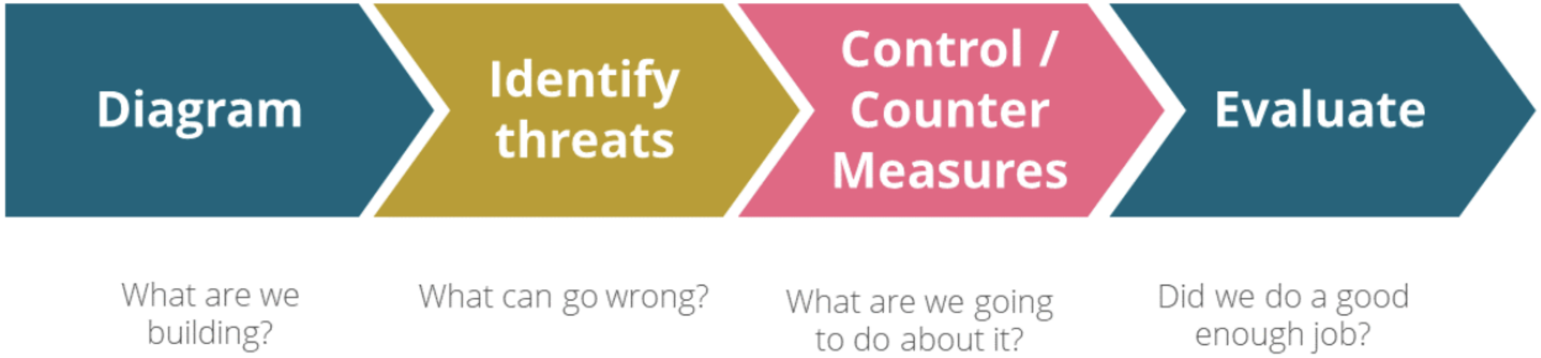
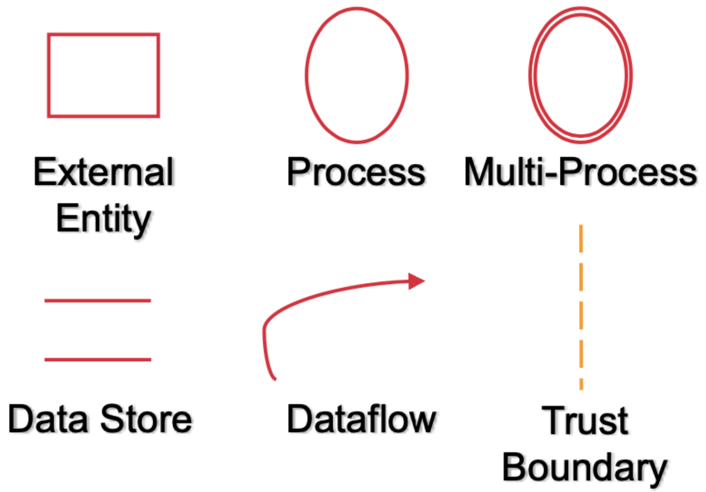


Diagram
Identify threats
Control / Counter Measures
Evaluate
What are we building?
What can go wrong?
What are we going to do about it?
Did we do a good enough job?

Context: do you understand your system?

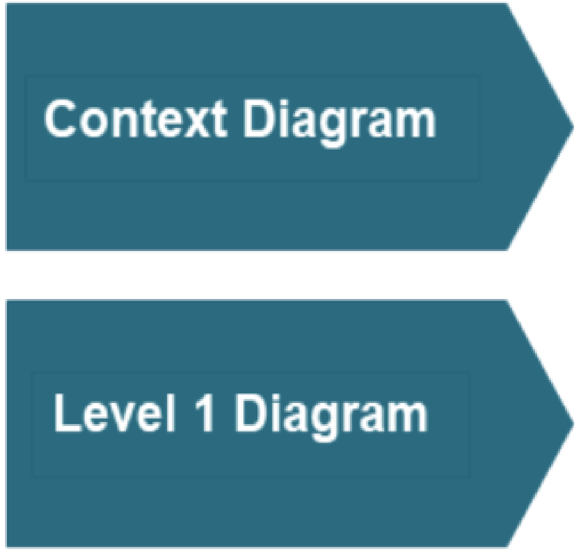
- Define the scope of TM
- Make sure everyone understands context, outcomes, how system works
- Know who works with or has access to software
- Common understanding of what is considered important (CIA-triade)
- Use data, sequence, state diagrams
- Identify attack surface
- Foundation of TM

Data flow diagrams (DFD)



- External entity: person or system interaction with application via an entry point (not in control)
- Process: tasks handling data within application (in control)
- Data store: locations where data is stored (not modified, i.e. DB)
- Data flow: Data movement within application, arrows
- Trust boundary: Identify locations where attackers might act, change of trust levels as data moves trhough the application

Types of DFD



Context Diagram

Very high level,
what interacts with my app?
Who interacts with my app?

Level 1 Diagram

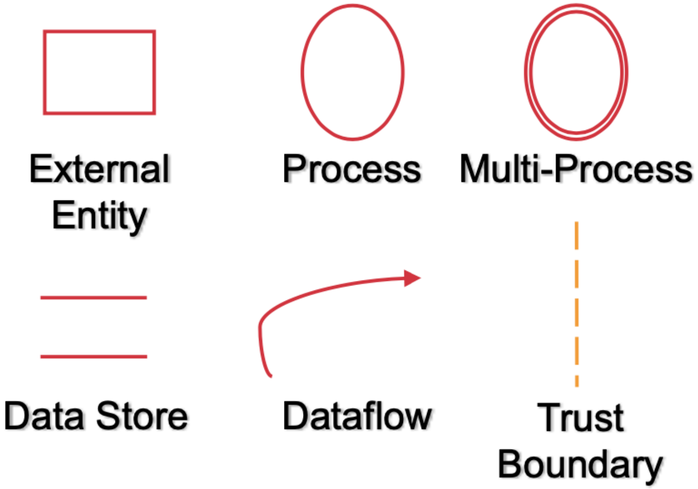
High level,
What are different components of my app?
How does data moves between process?
Where I need to check the level of trust netween the components?

Let`s make a context diagram: hotel management app

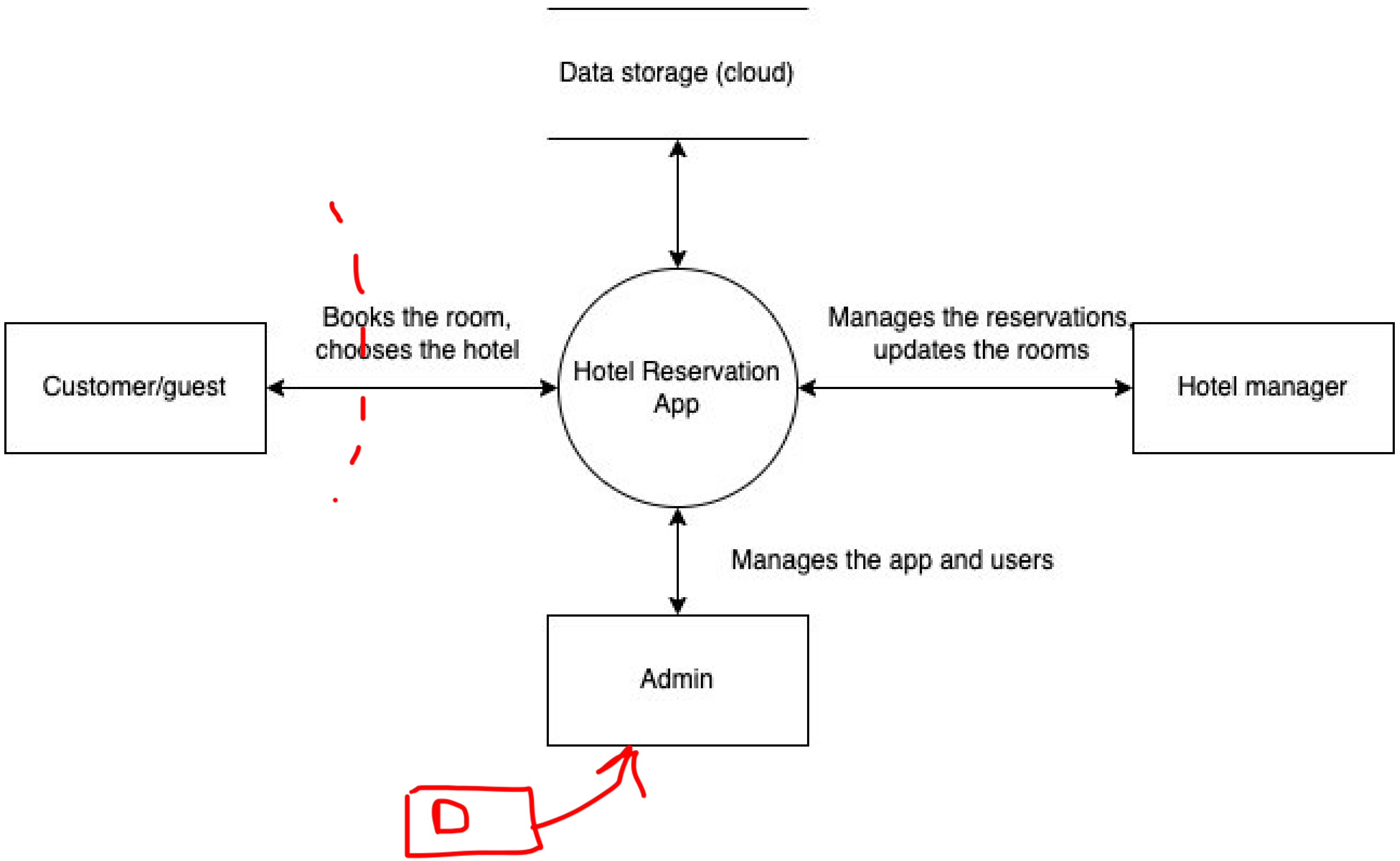
- Classic 3-tier web app:
  - Guests – login and book
  - Hotels – manage and update
  - Admin – manage the app
  - App
  - Data store - cloud
  - ….

Open draw.io

Or use pen and paper



Is it something like this?

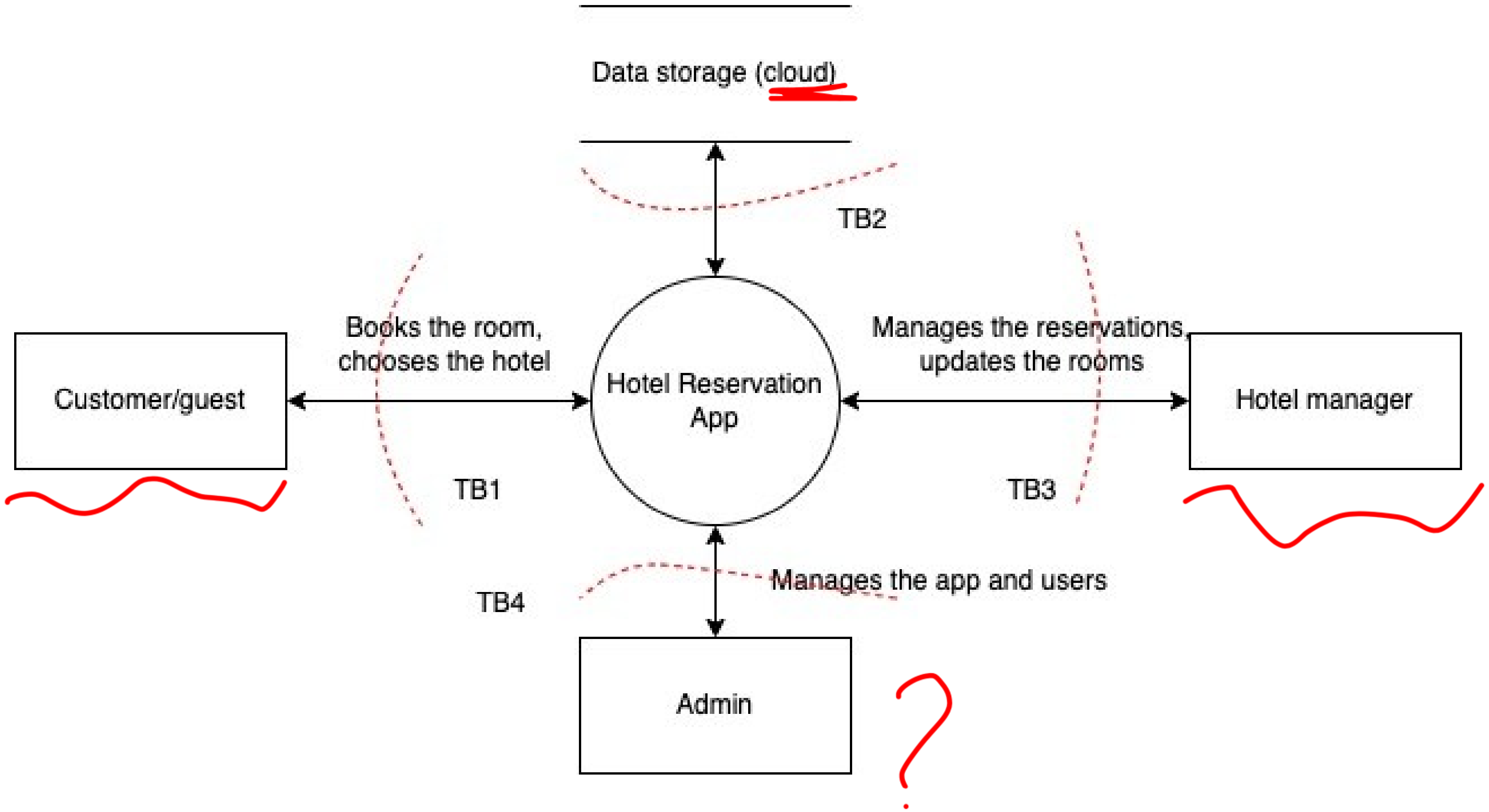

Data storage (cloud)
Books the room, chooses the hotel
Customer/guest
Hotel Reservation App
Manages the reservations, updates the rooms
Hotel manager
Manages the app and users
Admin
D

Let`s find trust boundaries

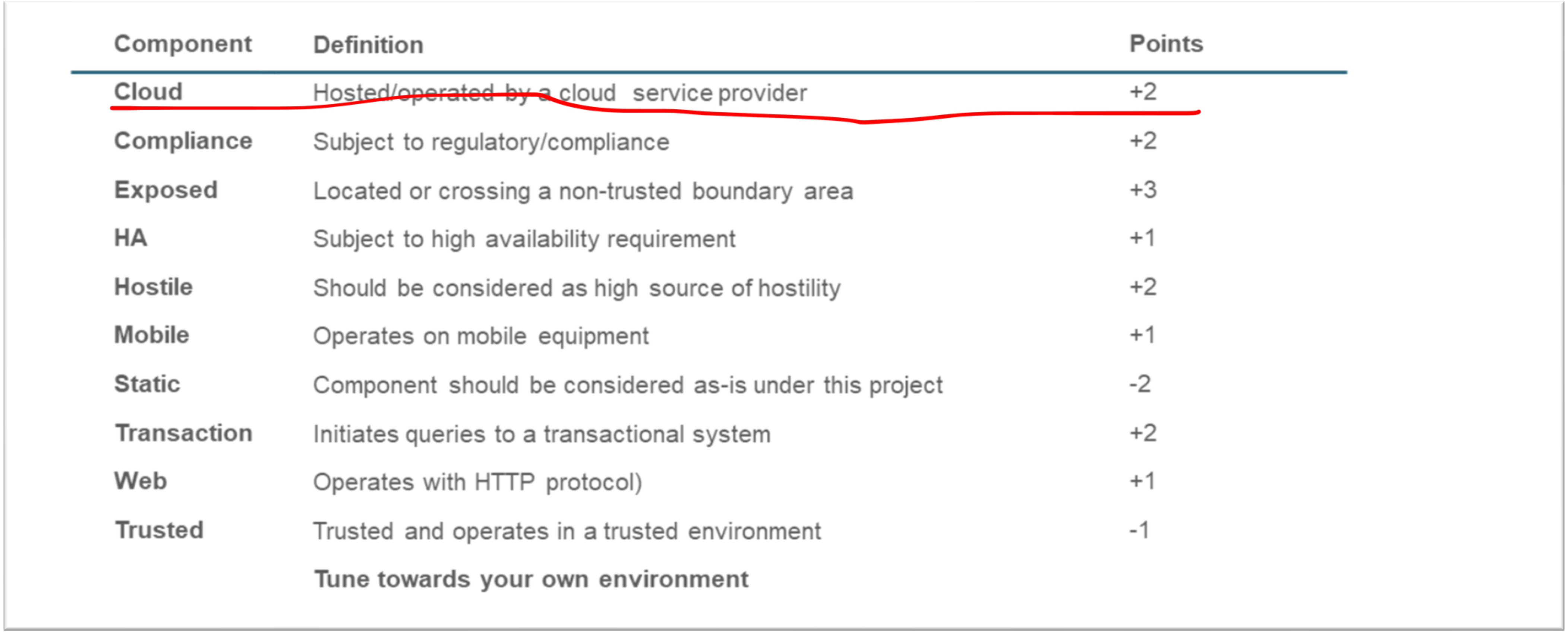
What are trust boundaries?

- Trust boundaries intersect data flows within a diagram
- Show where trust levels change
- Attack surface where an attacker can interject
- Examples: Machine boundaries, privilege boundaries, integrity boundaries
- Processes talking across a network always have a trust boundary

Is it something like this?



Priorities: choose 3 highest



Component	Definition	Points
Cloud	Hosted/operated by a cloud service provider	+2
Compliance	Subject to regulatory/compliance	+2
Exposed	Located or crossing a non-trusted boundary area	+3
HA	Subject to high availability requirement	+1
Hostile	Should be considered as high source of hostility	+2
Mobile	Operates on mobile equipment	+1
Static	Component should be considered as-is under this project	-2
Transaction	Initiates queries to a transactional system	+2
Web	Operates with HTTP protocol)	+1
Trusted	Trusted and operates in a trusted environment	-1
	Tune towards your own environment	

What are the top 3?

- 3 questions to each TB:
  - How far can we trust this external entity? What can go wrong ?
  - How far can we trust the communication protocol? What can go wrong?
  - How far can we trust our app? What can go wrong?

  And more…

- Who’s interested in app and data (threat agents)?
- What goals (assets)?
  What attack methods (how)?
- Any attack surfaces (trust boundaries) exposed?
- Any input/output (data flows) missing?

A best question

Is there anything keeping you up at night worrying about this system?

Practice

- Draw context Level 0 DFD diagram of your app
- Identify trust boundaries
- Identify the ones with highest priority
- Identify scenario for each untrusted trust boundary that scares you most (doomsday scenario).
    - What if the data of your application shows up on the dark web? (confidentiality)
    - What if your app is offline for a day? (availability)
    - What if data is randomly altered? (integrity)

Example: HR SaaS app

- App is used by enterprises (250+ employees) to store contracts, organigrams, 360 reviews, …
- Doomsday scenario's:
  - Data on darkweb (confidentiality): complete fiasco. Personal data of employees, social security numbers, wages. Almost as bad: unauthorized access to data (e.g. non-hr person that can access data of all their peers)
  - Outage (availability): not that bad: application is not mission critical for customers – 36h disaster recovery window
  - Data manipulation (integrity): relatively bad: employees could manipulate their ratings, … There's not a direct connection between this app and the payroll app, so no immediate financial impact.

Part 2

Component level (DFD Level 1)

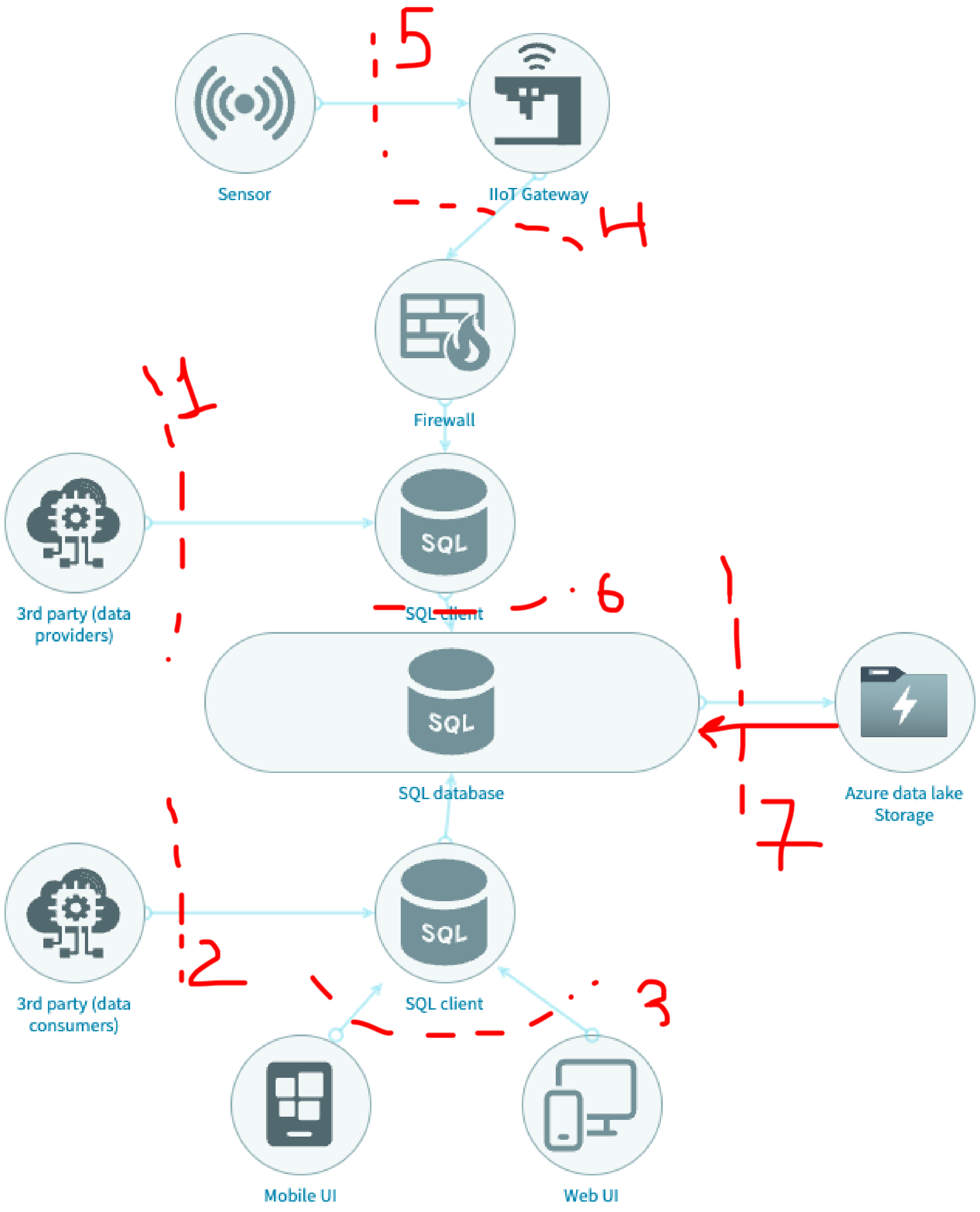
STRIDE

Risk evaluation

Example

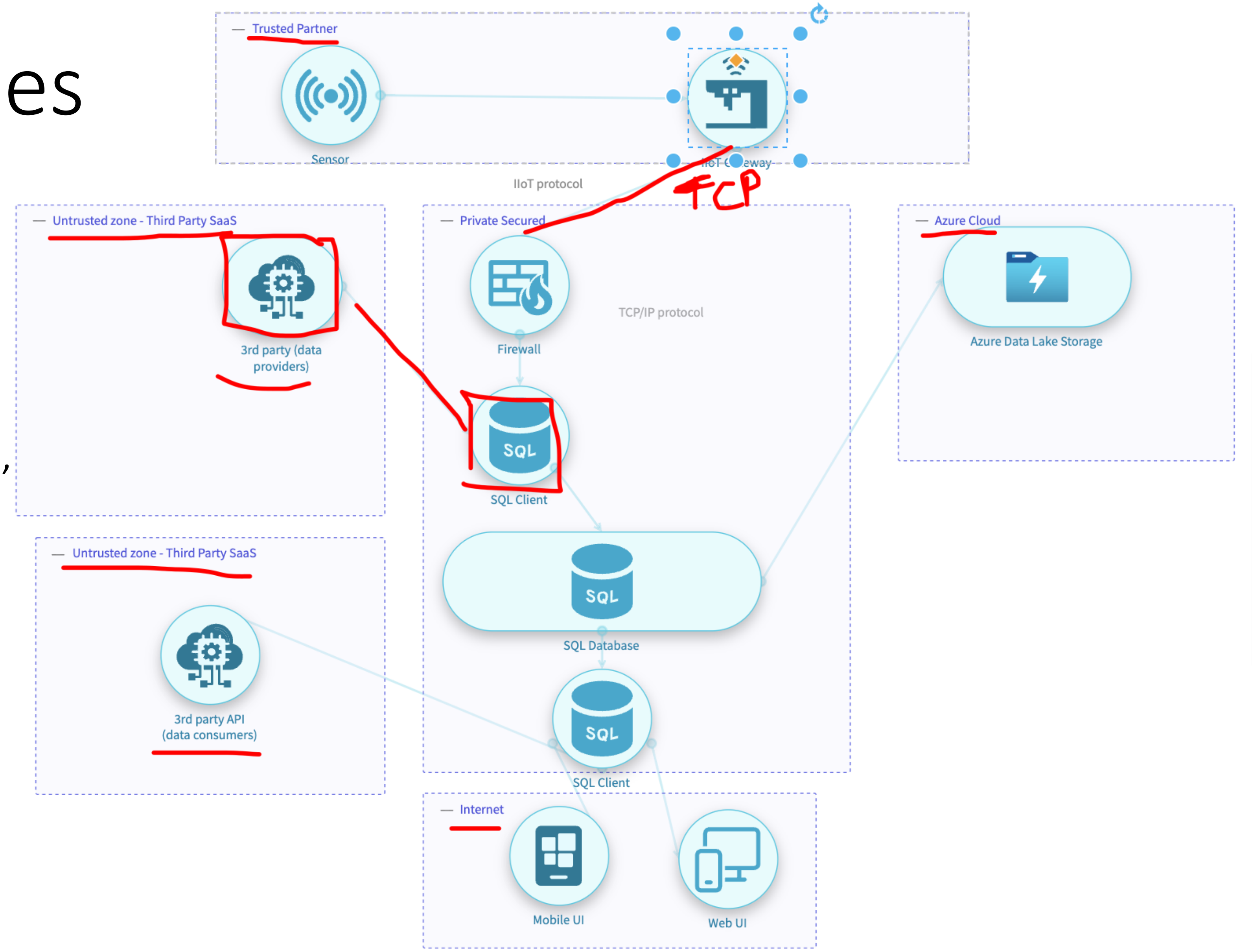
- SmartMeter digital quality management solution provides support to the digitalized quality tracking via IoT infrastructure in customer premises.
- The data (measurements) is collected from the sensors in customer premises and then stored and accessed in the database and in the cloud.
- Data is made available to the customers through the mobile app and web interface. This data is also available to external providers via JSON APIs
- There are two primary components in the system: sensors and backend.
- 2 types of devices are installed on the customer premise :
- Sensors : send measurement data to the gateway
- Gateways : receives measurement data from one or more sensor devices. Packages the data and sends it via unencrypted TCP to the backend. The communication is secured by a firewall.
- The main part of the solution is running at a datacenter. All data is stored on a SQL Database . The backend is also connected to Microsoft Azure. For reporting purposes, data is replicated from the SQL database to a Datalake in Azure.
- All the external links (data providers, cloud, data consumers, mobile app and web interface) go through https.

DFD Level 1 – find trust boundaries



DFD Level 1 – trust boundaries

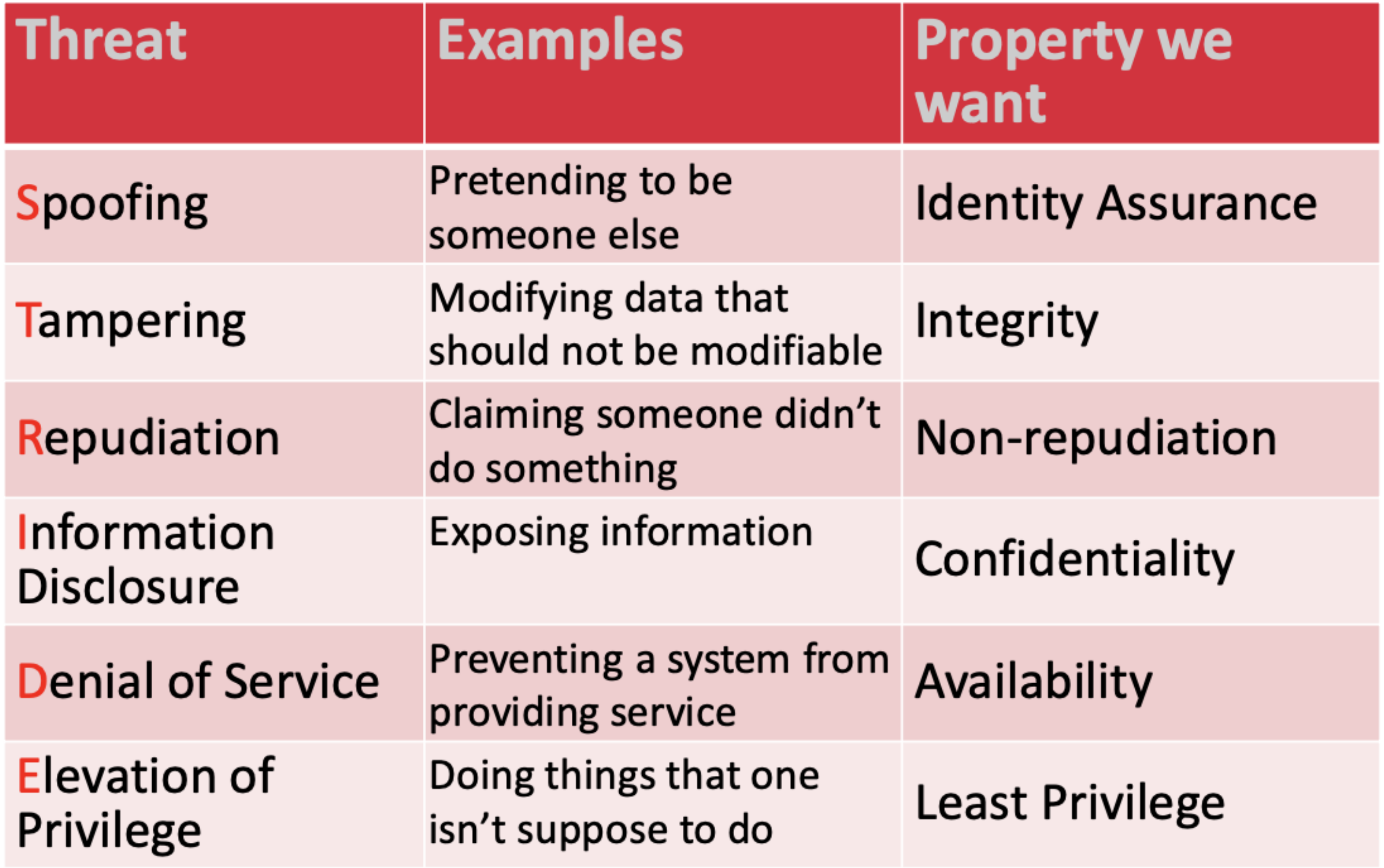
Threat1: spoofing of data due to leaked API credentials
TB: data providers –SQL client
Impact: high impact
Countermeasures: double check, Surveilance/monitoring, revoke/rotate, Storage
Component: SQL



STRIDE ANALYSIS

- Spoofing: Can attacker gain access using a false identity?
- Tampering: Can attacker modify data in the application?
- Repudiation: If attacker denies doing something, can we prove it?
- Information disclosure: Can attacker get access to sensitive data?
- Denial of Service: Can attacker crash or reduce availability of the application?
- Elevation of privilege: Can attacker take identity of a privileged user?

STRIDE Framework – Data Flow



Threat	Examples	Property we want
Spoofing	Pretending to be someone else	Identity Assurance
Tampering	Modifying data that should not be modifiable	Integrity
Repudiation	Claiming someone didn't do something	Non-repudiation
Information Disclosure	Exposing information	Confidentiality
Denial of Service	Preventing a system from providing service	Availability
Elevation of Privilege	Doing things that one isn't suppose to do	Least Privilege

	HOW TO CONTROL
SPOOFING	Authentication based on key exchange Decide on single-factor, two-factor, or multi-factor authentication Offload authentication to another provider Restrict authentication to certain IP ranges or locations
TAMPERING	Data protected from tampering with cryptographic integrity mechanisms Only enumerated authorized users may modify data
REPUDIATION	Maintain logs Digital signature
INFORMATION DISCLOSURE	Data in files / database will only be available to authorized users Name / existence of database will only be exposed to authorized users Content and existence of communication between Alice and Bob will only be exposed to these authorized users
DENIAL OF SERVICE	Rate limiting or throttling access to a service Real-time monitoring of log files and other resources to note sudden changes
ELEVATION OF PRIVILEGE	System has a central authorization engine Authorization controls stored with item being controlled using ACLs System limits who can write data to higher integrity level System uses roles / accounts or permissions to manage access

Threat table – current status

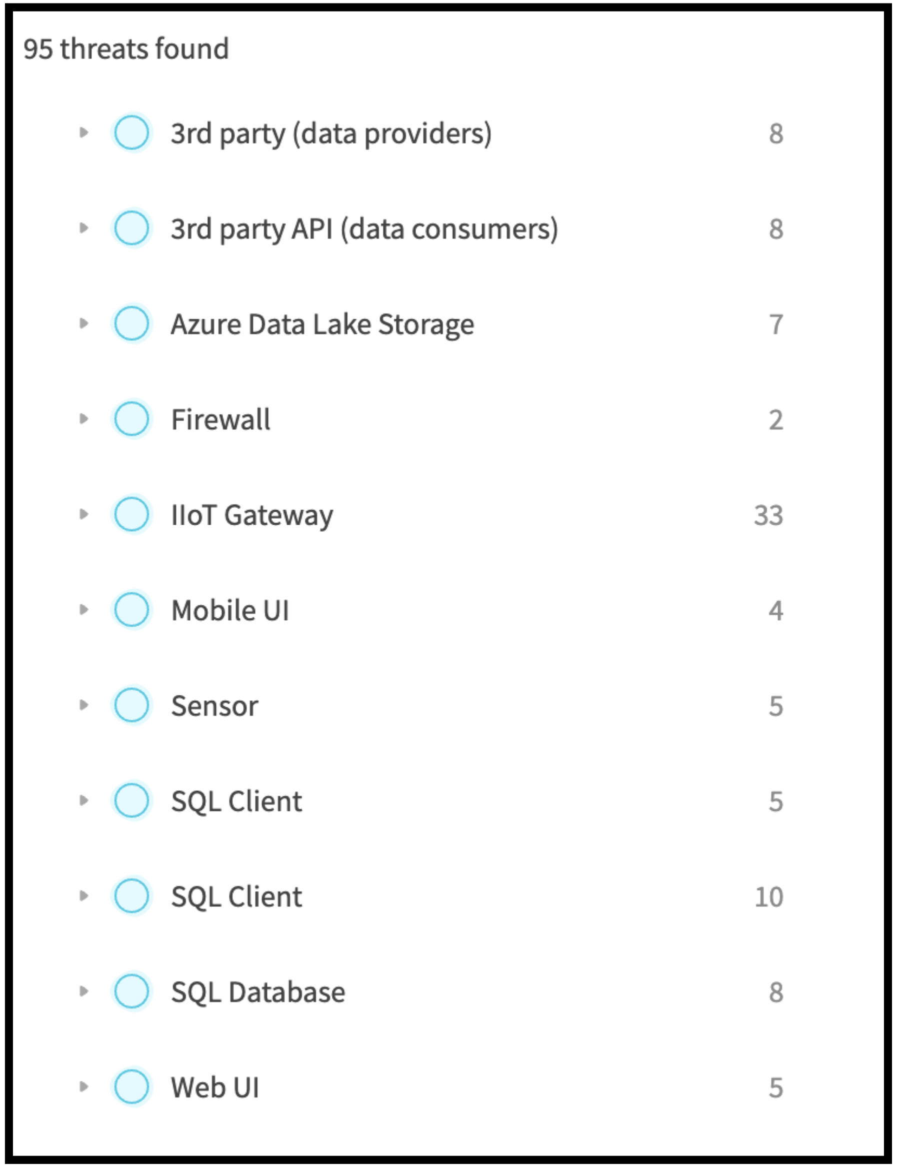
	External entity	External entity	Link	Link	Component	Component
TB1	Mitigations: What controls are there?	Vulnerabilities: What can go wrong?	Mitigations	Vulnerabilities	Mitigations	Vulnerabilities
S	Firewall	V1. No checking of data source	TCP/IP	V1. Unencrypted communication		
T						
R						
I						
D						
E						

Threat found:

RISK	HIGH?
THREAT	There is no data source check on a firewall, the data could be spoofed or tampered by an attacker
IMPACT	Incorrect data in the system
COUNTER MEASURE	IP RANGE of CUSTOMERS
COMPONENT	FIREWALL

RISK	HIGH?
THREAT	There is encryption in the communication protocol, the data could be spoofed or tampered by an attacker
IMPACT	Incorrect data in the system
COUNTER MEASURE	ENCRYPT
COMPONENT	COMMUNICATION PROTOCOL

Example threat list



95 threats found

3rd party (data providers)	8
3rd party API (data consumers)	8
Azure Data Lake Storage	7
Firewall	2
IIoT Gateway	33
Mobile UI	4
Sensor	5
SQL Client	5
SQL Client	10
SQL Database	8
Web UI	5

Mitigations

1. Leave as-is
2. Remove from product
3. Remedy with technology countermeasure
4. Warn user

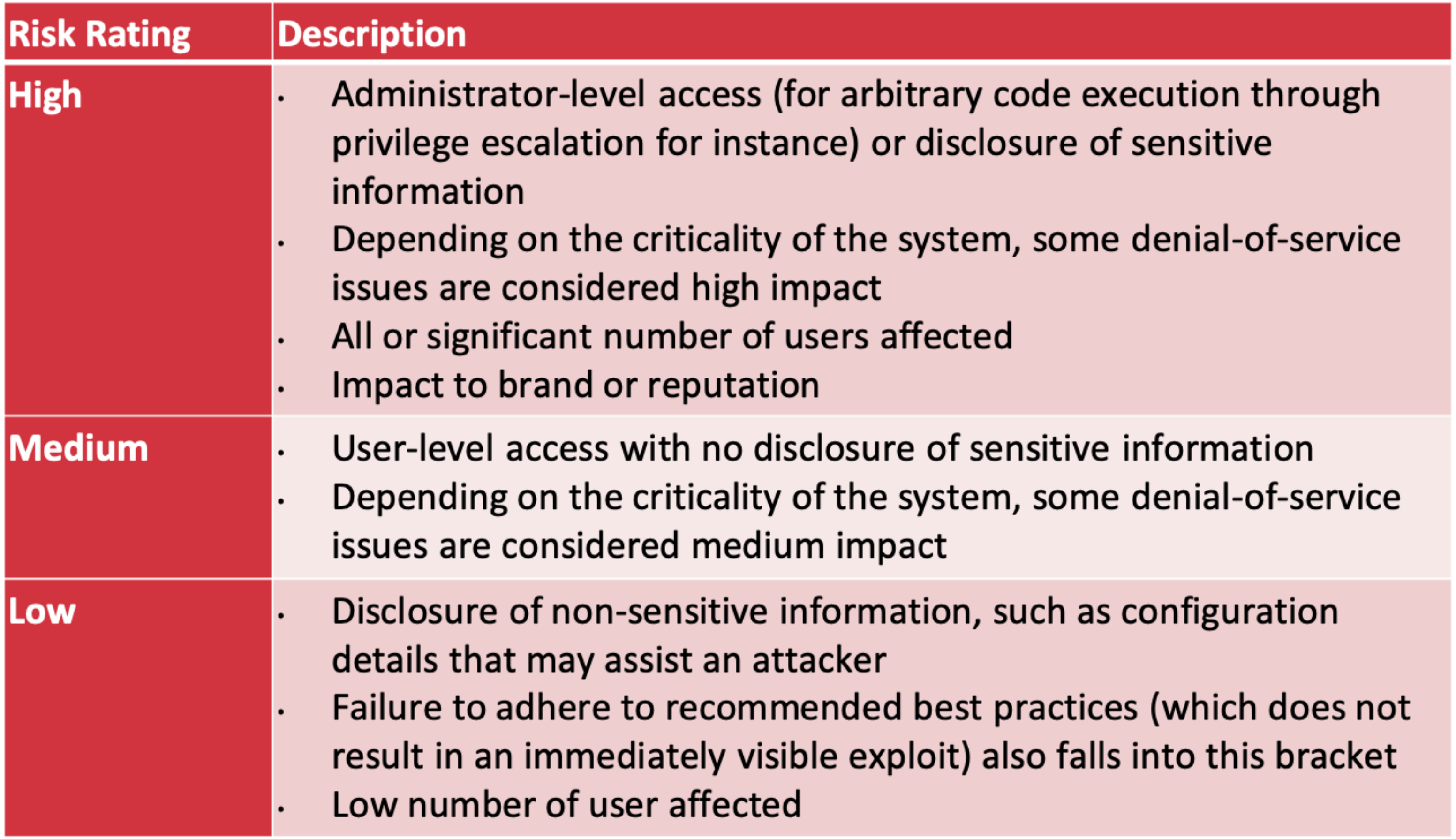
Evaluate risks

- Ease of exploitation
- Business impact
- High, medium, low

Ease of exploitation

Risk Rating	Description
High	- Tools and exploits are readily available on the Internet or other locations - Exploitation requires no specialized knowledge of the system and little or no programming skills - Anonymous users can exploit the issue
Medium	- Tools and exploits are available but need to be modified to work successfully - Exploitation requires basic knowledge of the system and may require some programming skills - User-level access may be a pre-condition
Low	- Working tools or exploits are not readily available - Exploitation requires in-depth knowledge of the system and/or may require strong programming skills - User-level (or perhaps higher privilege) access may be one of a number of pre-conditions

Business impact



Risk Rating	Description
High	• Administrator-level access (for arbitrary code execution through privilege escalation for instance) or disclosure of sensitive information • Depending on the criticality of the system, some denial-of-service issues are considered high impact • All or significant number of users affected • Impact to brand or reputation
Medium	• User-level access with no disclosure of sensitive information • Depending on the criticality of the system, some denial-of-service issues are considered medium impact
Low	• Disclosure of non-sensitive information, such as configuration details that may assist an attacker • Failure to adhere to recommended best practices (which does not result in an immediately visible exploit) also falls into this bracket • Low number of user affected

Practice

- Draw DFD Level 1 (component diagram)
- Identify trust boundaries
- Select 1 trust boundary to do STRIDE analysis for external entity, process and communication link (so you will have 3X6 situations).
- Specify the threats of this trust boundary (at least 10 of them)
- Evaluate risks for each threat – ease of exploitation and business impact
- Specify mitigations for the 3 highest risks

Start today!

- Start with secure design as goal
- Ask the “what if” questions
- Understand bigger picture

Useful links

- https://github.com/hysnsec/awesome-threat-modelling?tab=readme-ov-file
- OWASP Threat modeling manifesto
- https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html
- https://safecode.org/
- https://ieeecs-media.computer.org/media/technical-activities/CYBSI/docs/Top-10-Flaws.pdf
- https://github.com/rhurlbut/CodeMash2019/blob/master/Robert-Hurlbut-CodeMash2019-Threat-Modeling-Workshop-20190108.pdf
- https://www.toreon.com/threatmodeling/

END of this workshop

But how do I do it in practice?

Sirris is working on a startup kit,

To help you initiate TM process in your company

And review of TM tools.