IT-Sicherheit

Kapitel 6: Secure Software Engineering

Teil 1

- Vorgehensmodell
- Analyse der Sicherheits-Anforderungen
- Sicherheitsarchitektur und- Design
- Tools zur Sicherheitsanalyse

Prof. Dr. Reiner Hüttl, TH Rosenheim, © 2021, 15.03.2021

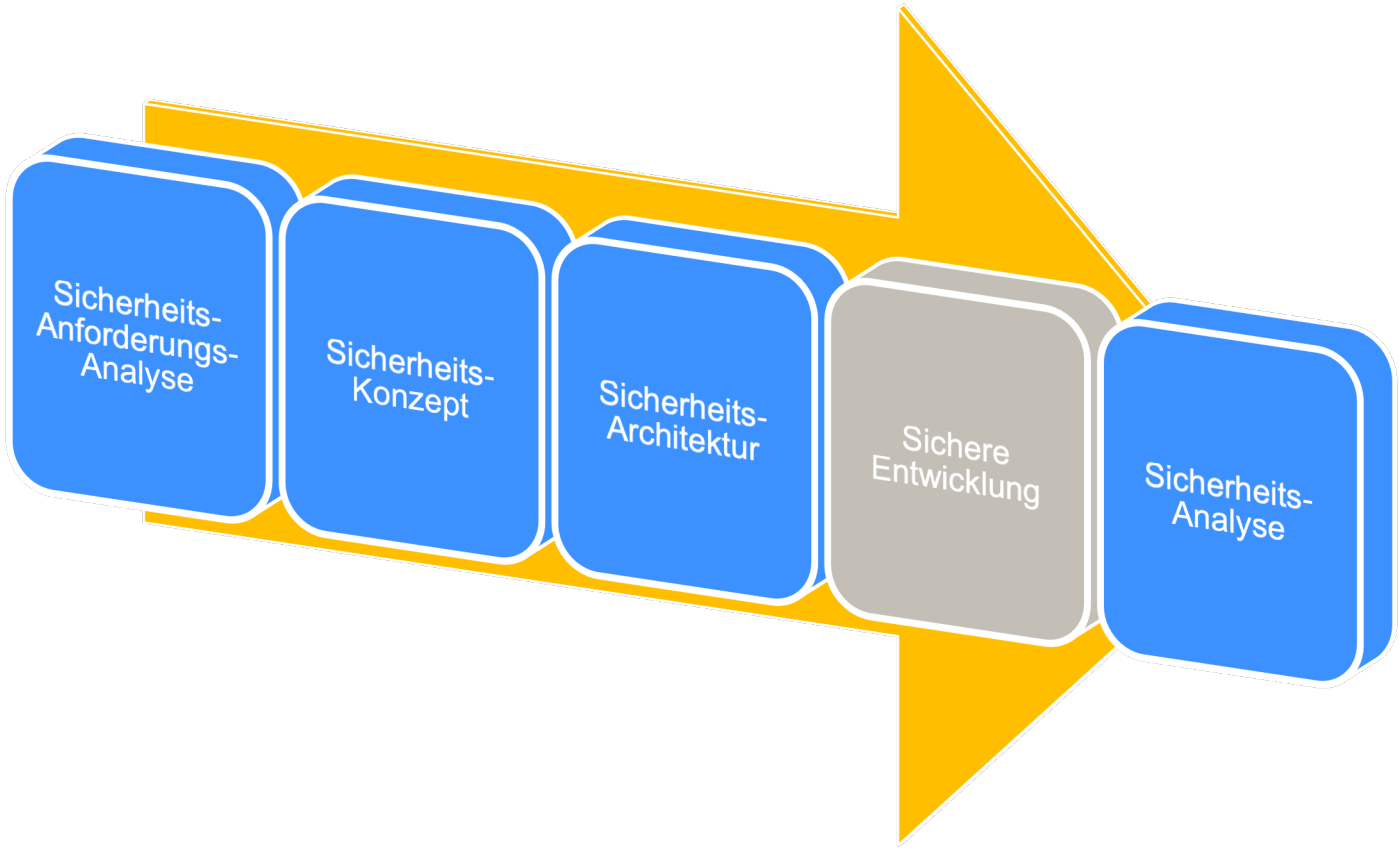
Worum geht es?

- Wie bekommen wir Sicherheit in den Software Engineering Prozess?
- Was sollte man vor der Implementierung bezüglich Sicherheit beachten?
- Welche Designprinzipen gibt es für Sicherheit?
- Wie kann ich Sicherheit in meinem IT-System überprüfen?

Wozu Secure Software Engineering?

- Unsichere Software kann böse Überraschungen liefern

Die Phasen von Secure Software Engineering



Softwareentwicklung auf dem Software-Fließband

- Die Produktqualität wird automatisiert und holistisch bestimmt

Omnipräsente Feedback-Devices

CI/CD Pipeline

Holistische Qualitätsanalysen

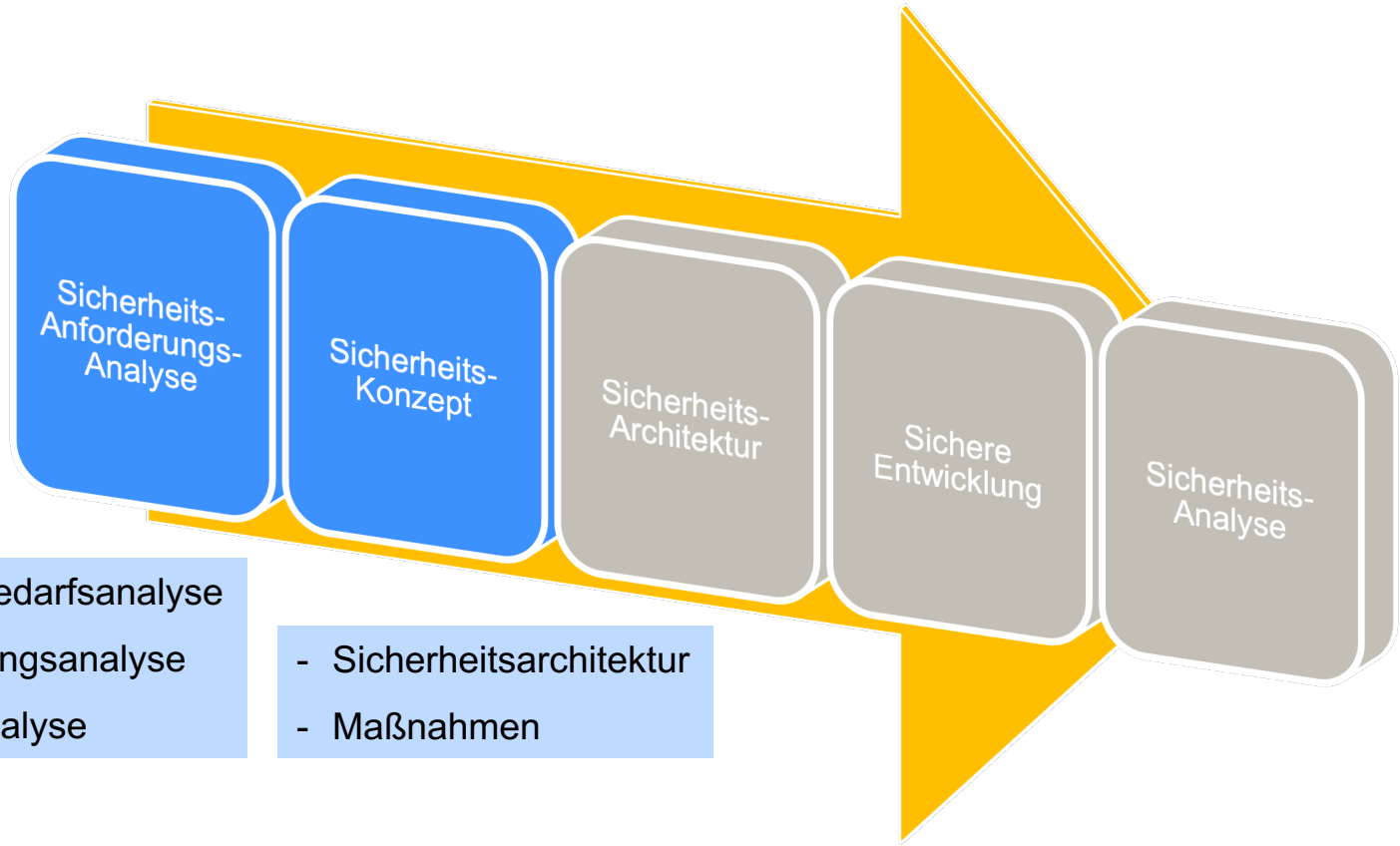
Funktionalität?
Wartbarkeit?
Security?
Compliance?
Performance?
...

Integrierte Software

Fertige Software

Code-Änderung

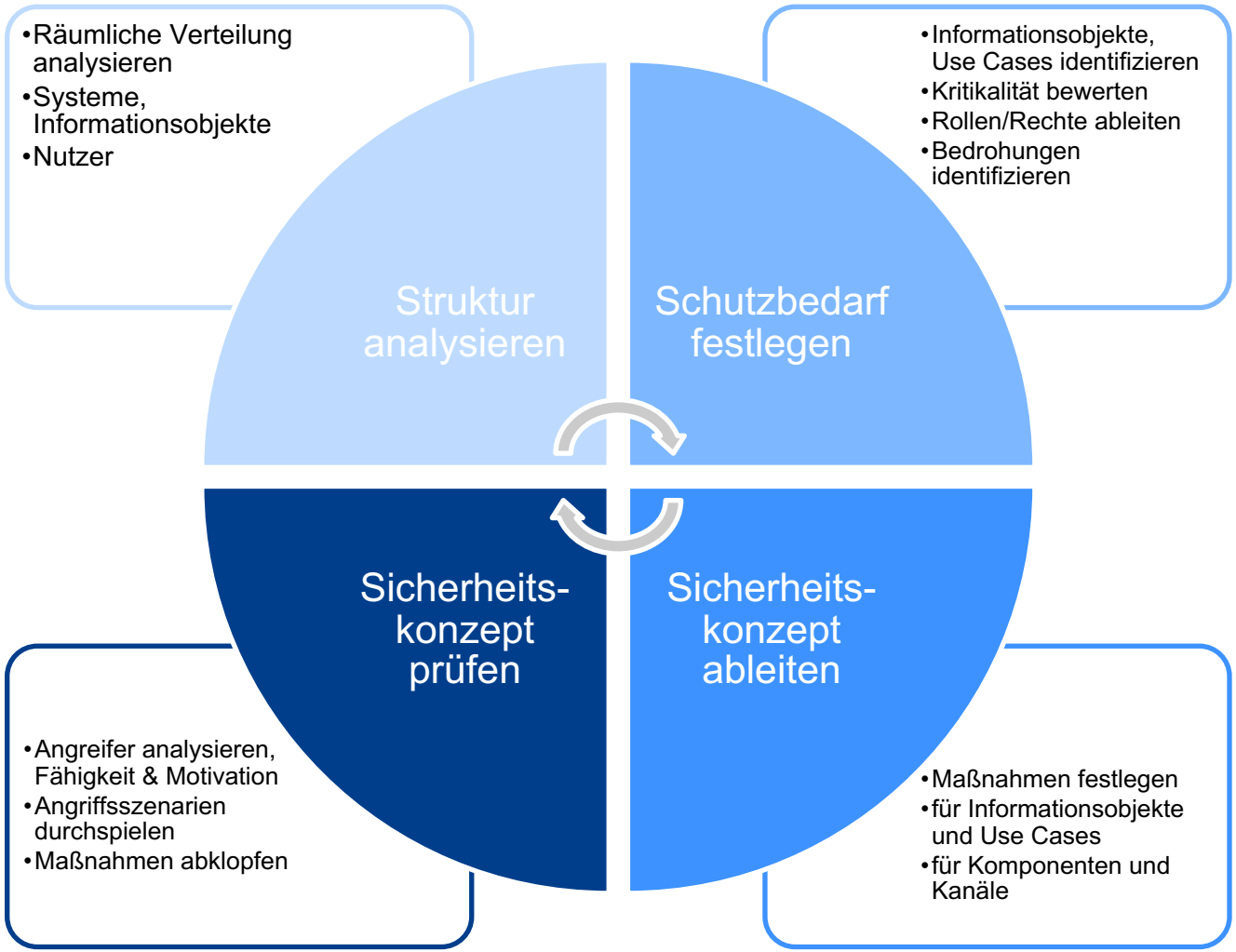
Die Sicherheits-Analyse Phase



- Schutzbedarfsanalyse
- Bedrohungsanalyse
- Risikoanalyse

- Sicherheitsarchitektur
- Maßnahmen

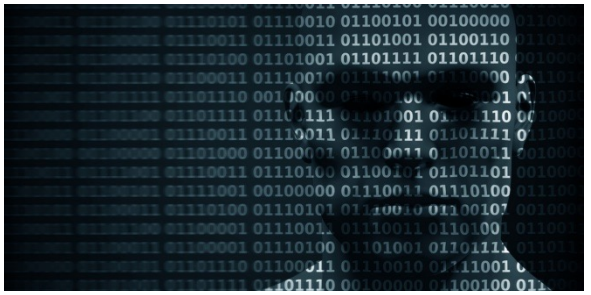
Ein Vorgehensmodell für Sicherheits-Analysen



Der Schritt 2 im Vorgehensmodell: Schutzbedarf festlegen

- Kritische Informationsobjekte identifizieren
  - Bewertung bezüglich Sicherheitsziele
  - Welcher Schaden droht bei Verletzung von Sicherheitszielen?
- Schutzbedarf der Use-Cases bewerten
  - Bei welchen Use Cases droht Schaden bei Verletzung von Sicherheits-Zielen?
  - Auch technische Use-Cases betrachten (z.B. Zertifikats-Management, System-Administration, Berechtigungsvergabe)
- Rollen- und Rechtevergabe im System
  - Welche Anwender/Rollen gibt es?
  - Wer darf was?
  - Prinzipien festlegen („need to know“, „segregation of duties“, …)
- Bedrohungen identifizieren und analysieren
  - Threat Modeling
  - Risikoanalyse

Ein Modell zur Bedrohungsanalyse



Microsoft Threat Model: STRIDE

- **S**poofing (Manipulation und Täuschung)
  - Users should not be able to become any other user or assume the attributes of another user
- **T**ampering (Verfälschung)
  - Data tampering involves the malicious modification of persistent data and data over networks.
- **R**epudiation (Verleugnung)
  - Users may dispute transactions if there is insufficient auditing or recordkeeping of their activity

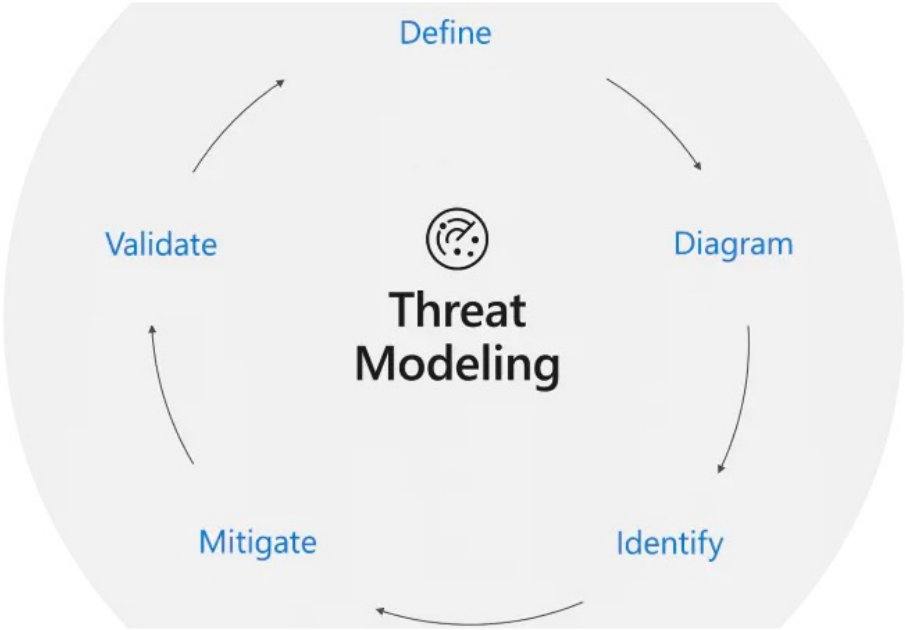
https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)

STRIDE (Teil 2)

Microsoft Threat Model: STRIDE

- **I**nformation Disclosure (Enthüllung und Abhören)
  - The exposure of information to individuals who are not supposed to have access to it
- **D**enial of Service (Überflutung)
  - Deny service to valid users—for example, by making a Web server temporarily unavailable or unusable.
- **E**levation of Privilege (Erschleichen von Berechtigungen)
  - An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system

Threat Modeling nach Microsoft



There are five major threat modeling steps:

- Defining security requirements.
- Creating an application diagram.
- Identifying threats.
- Mitigating threats.
- Validating that threats have been mitigated.

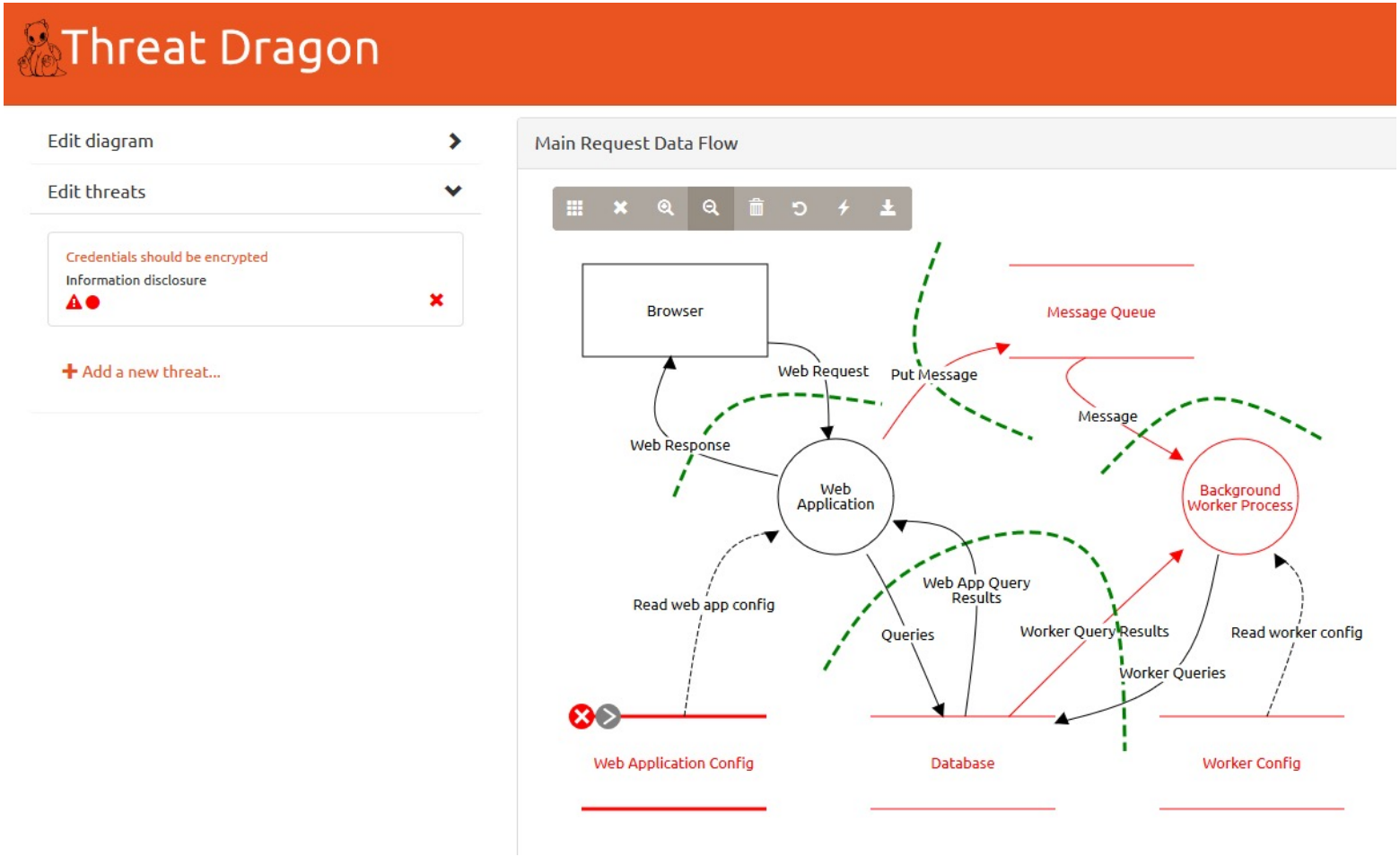
Quelle: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

Microsoft Thread Modeling Tool https://aka.ms/threatmodelingtool

- Alternative Vorgehensweisen zur Bedrohungsanalyse
  - Misuse cases
  - Attack Trees
  - Bedrohungskataloge

Weitere Informationen in:
Matthias Rohr: Sicherheit von Webanwendungen in der Praxis, Springer Vieweg, 2018 **(E-Book)**

Beispiel für Threat Modeling: Application Diagram



OWASP Threat Dragon, https://threatdragon.org/#/

https://threatdragon.org/#/threatmodel/mike-goodwin/owasp-threat-dragon-demo/master/Demo%20Threat%20Model/diagram/0

Beispiel für Threat Modeling: Threats and Mitigations

Threat	Type	Mitigation
Unauthorized request to DB	I	All queries to be authenticated
DB Credential Theft	I	Use FW to restrict access to DB to only background Worker IP
Message Tampering in Message queue	T	Sign all messages
Fake massages in queue	S	Implement authentication on queue
Generate malicious messages that Background Worker cannot process	D	Validate content of messages before processing, reject messages with invalid content, log the rejection, do not log the malicious content
Brute forcing of Web Application Login	E	Slowdown login attempt after unsuccessful login, 2FA for admin accounts
Sniffing of Web requests	I	Https Encryption of all requests
SQL injection	T	Input validation
Undocumented change of Web App Config	R	Auditing all changes in Web App Config, access control to Web App Config

Risikoanalyse

- Eine umfängliche Risiko-/Bedrohungsanalyse ist aufwendig
  - Oft ist die Bereitschaft beim Kunden/Auftraggeber nicht vorhanden
  - → Führe ein pragmatische Risikoanalyse durch
- Konzentriere dich auf die wichtigsten Risiken
- Lass dich von der Datenkritikalität und den Schnittstellen leiten
- Risiken müssen durch die Verantwortlichen bewertet werden (ISO, DPO, Product Owner, Management)
- Stelle Transparenz über die Bewertung der Risiken her
  - Review durch ISO/DPO

Risikoanalyse am Beispiel Logging

Logstash **Kibana**

- Wir betrachten als Beispiel das Logging in einer Cloud Anwendung

- Security Goals

  - The root cause of incidents or faulty platform or application behavior can be adequately analyzed and identified.
  - Required log data and analysis tools are available and correspond to the actual state of the system at the relevant time.
  - The technical logs are secured from unauthorized access and manipulation.

Risiken beim Logging

- **R-1: Missing log data.** An incident cannot be sufficiently analyzed because relevant log information for the required period of time has not been collected, e.g. due to a misconfiguration/failure of the log stack or according infrastructure components.
- **R-2: Loss of log data.** Log information gets lost, e.g. due to a failure of the log storage.
- **R-3: Manipulation of logs.** The root cause of an incident can be hidden or obscured by modification or deletion of log data.
- **R-4: No access to log data.** Relevant log data cannot be viewed when required due to blocked access, e.g. missing credentials
- **R-5: Disclosure of sensitive log information.** Information written to log files can give valuable guidance to an attacker or expose sensitive user data
- **R-6: Violation of deletion obligation.** To store log files longer than the allowed retention period violates compliance (e.g. GDPR)

⚡ Verfügbarkeit

⚡ Integrität

⚡ Vertraulichkeit

Risk-Control-Matrix für Logging

System Component	Risk	Risk name	Mitigating measures
Logging	R-.1	Missing log data	- all logs are collected and stored in a central managed log stack - log configuration is maintained by DevOps experts - regular review of all critical assets for their correctness and currency - mechanism to ensure that all required logs are captured (e.g. via documented search in logging system, configuration rule/policy)
Logging	R-2	Loss of log data	- backup of log data by AWS - storage of log data provided by AWS in a managed ELK stack - retention of 30 days - independent monitoring of logging software with alerting in case of failure
Logging	R-3	Manipulation of logs	- log data secured by AWS - access control via IAM - measures for integrity - audit the access to log data
Logging	R-4	No access to log data	- availability is provided by AWS
Logging	R-5	Disclosure of sensitive log information	- isolation of application log data (separate storage and access control for different applications/tenants) - role-based access control to logs - encryption of data at rest, decryption key only available to application owner - transport of log data is secured with minimum TLS 1.2
Logging	R-6	Violation of deletion obligation	- complete deletion of log data immediately after end of retention period - there are no local copies / snapshots of log data (enforced by policy) - deletion process according to GDPR and security needs

Wie findet man technische Maßnahmen für die Sicherheitsziele?

Sicherheitsziele

Nicht-Abstreitbarkeit
Jede durchgeführte Aktion ist nachweisbar genau so passiert

Integrität
Keine unbefugte Manipulation von Daten und Funktionen

Vertraulichkeit
Keiner erhält unerlaubten Zugriff auf Daten, Nachrichten und Funktionen.

Verfügbarkeit
Daten und Funktionen sind stets verfügbar, wenn sie benötigt werden und für diejenigen, die sie benötigen.

Authentizität

Echtheit von Daten, Zurechenbarkeit von Nachrichten

Das Ergebnis der Sicherheitsanalyse ist ein Sicherheitskonzept