Threagile
Agile Threat Modeling

Threat Model Report
Secure IT Testmodell

10 July 2025

test

Table of Contents

Management Summary

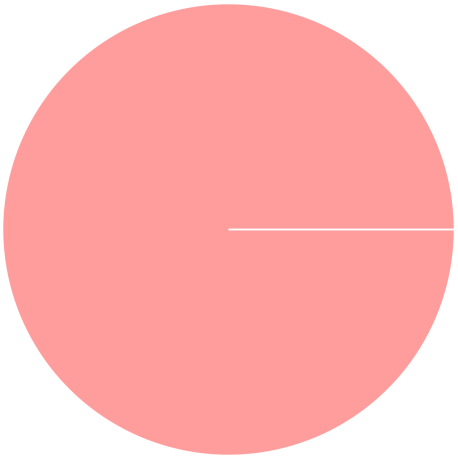
Threagile toolkit was used to model the architecture of "Secure IT Testmodell" and derive risks by analyzing the components and data flows. The risks identified during this analysis are shown in the following chapters. Identified risks during threat modeling do not necessarily mean that the vulnerability associated with this risk actually exists: it is more to be seen as a list of potential risks and threats, which should be individually reviewed and reduced by removing false positives. For the remaining risks it should be checked in the design and implementation of "Secure IT Testmodell" whether the mitigation advices have been applied or not.

Each risk finding references a chapter of the OWASP ASVS (Application Security Verification Standard) audit checklist. The OWASP ASVS checklist should be considered as an inspiration by architects and developers to further harden the application in a Defense-in-Depth approach. Additionally, for each risk finding a link towards a matching OWASP Cheat Sheet or similar with technical details about how to implement a mitigation is given.

In total **9 initial risks** in **7 categories** have been identified during the threat modeling process:

0 critical risk
0 high risk
3 elevated risk
6 medium risk
0 low risk

9 unchecked
0 in discussion
0 accepted
0 in progress
0 *mitigated*
0 *false positive*



Impact Analysis of 9 Initial Risks in 7 Categories

The most prevalent impacts of the **9 initial risks** (distributed over **7 risk categories**) are (taking the severity ratings into account and using the highest for each category):
Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated: **Cross-Site Scripting (XSS)**: 2 Initial Risks - Exploitation likelihood is *Likely* with *Medium* impact.
If this risk remains unmitigated, attackers might be able to access individual victim sessions and steal or modify user data.

Elevated: **SQL/NoSQL-Injection**: 1 Initial Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to modify SQL/NoSQL queries to steal and modify data and eventually further escalate towards a deeper system penetration via code executions.

Medium: **Missing Build Infrastructure**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model due to critical build infrastructure components missing in the model.

Medium: **Missing Cloud Hardening**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *High* impact.
If this risk is unmitigated, attackers might access cloud components in an unintended way.

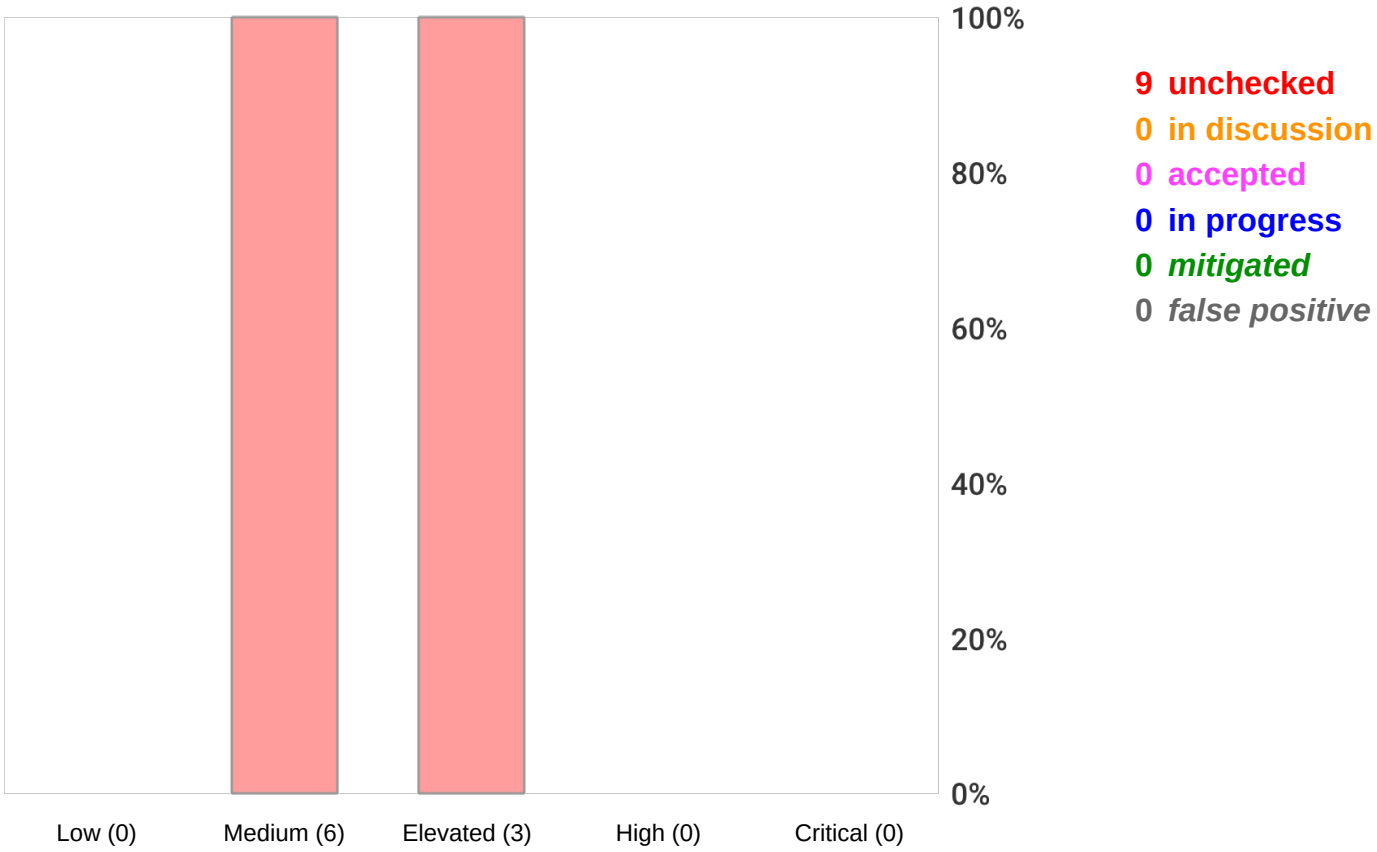
Medium: **Missing Hardening**: 2 Initial Risks - Exploitation likelihood is *Likely* with *Low* impact.
If this risk remains unmitigated, attackers might be able to easier attack high-value targets.

Medium: **Missing Identity Store**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model in the identity provider/store that is currently missing in the model.

Medium: **Missing Vault (Secret Storage)**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to easier steal config secrets (like credentials, private keys, client certificates, etc.) once a vulnerability to access files is present and exploited.

Risk Mitigation

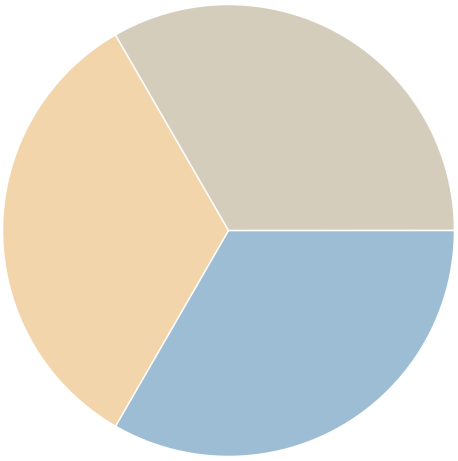
The following chart gives a high-level overview of the risk tracking status (including mitigated risks):



After removal of risks with status *mitigated* and *false positive* the following **9 remain unmitigated**:

0 unmitigated critical risk
0 unmitigated high risk
3 unmitigated elevated risk
6 unmitigated medium risk
0 unmitigated low risk

0 business side related
3 architecture related
3 development related
3 operations related



Impact Analysis of 9 Remaining Risks in 7 Categories

The most prevalent impacts of the **9 remaining risks** (distributed over **7 risk categories**) are (taking the severity ratings into account and using the highest for each category):
Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated: **Cross-Site Scripting (XSS)**: 2 Remaining Risks - Exploitation likelihood is *Likely* with *Medium* impact.
If this risk remains unmitigated, attackers might be able to access individual victim sessions and steal or modify user data.

Elevated: **SQL/NoSQL-Injection**: 1 Remaining Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to modify SQL/NoSQL queries to steal and modify data and eventually further escalate towards a deeper system penetration via code executions.

Medium: **Missing Build Infrastructure**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model due to critical build infrastructure components missing in the model.

Medium: **Missing Cloud Hardening**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *High* impact.
If this risk is unmitigated, attackers might access cloud components in an unintended way.

Medium: **Missing Hardening**: 2 Remaining Risks - Exploitation likelihood is *Likely* with *Low* impact.
If this risk remains unmitigated, attackers might be able to easier attack high-value targets.

Medium: **Missing Identity Store**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model in the identity provider/store that is currently missing in the model.

Medium: **Missing Vault (Secret Storage)**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to easier steal config secrets (like credentials, private keys, client certificates, etc.) once a vulnerability to access files is present and exploited.

Application Overview

Business Criticality

The overall business criticality of "Secure IT Testmodell" was rated as:

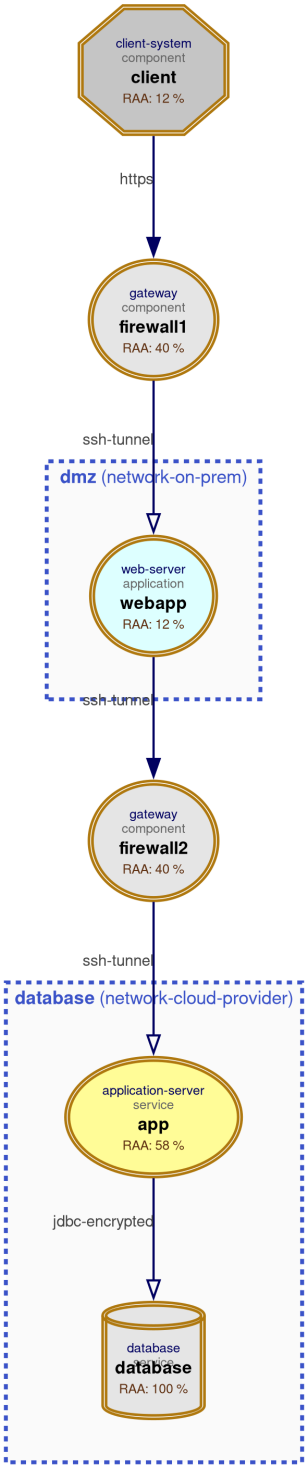
(archive | operational | **IMPORTANT** | critical | mission-critical)

Business Overview

Technical Overview

Data-Flow Diagram

The following diagram was generated by Threagile based on the model input and gives a high-level overview of the data-flow between technical assets. The RAA value is the calculated *Relative Attacker Attractiveness* in percent. For a full high-resolution version of this diagram please refer to the PNG image file alongside this report.



Security Requirements

This chapter lists the custom security requirements which have been defined for the modeled target.

This list is not complete and regulatory or law relevant security requirements have to be taken into account as well. Also custom individual security requirements might exist for the project.

Abuse Cases

This chapter lists the custom abuse cases which have been defined for the modeled target.

This list is not complete and regulatory or law relevant abuse cases have to be taken into account as well. Also custom individual abuse cases might exist for the project.

Tag Listing

This chapter lists what tags are used by which elements.

pii
user-data

STRIDE Classification of Identified Risks

This chapter clusters and classifies the risks by STRIDE categories: In total **9 potential risks** have been identified during the threat modeling process of which **1 in the Spoofing** category, **7 in the Tampering** category, **0 in the Repudiation** category, **1 in the Information Disclosure** category, **0 in the Denial of Service** category, and **0 in the Elevation of Privilege** category.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Spoofing

Medium: **Missing Identity Store**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

The modeled architecture does not contain an identity store, which might be the risk of a model missing critical assets (and thus not seeing their risks).

Tampering

Elevated: **Cross-Site Scripting (XSS)**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Medium* impact.

For each web application Cross-Site Scripting (XSS) risks might arise. In terms of the overall risk level take other applications running on the same domain into account as well.

Elevated: **SQL/NoSQL-Injection**: 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

When a database is accessed via database access protocols SQL/NoSQL-Injection risks might arise. The risk rating depends on the sensitivity technical asset itself and of the data assets processed or stored.

Medium: **Missing Build Infrastructure**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

The modeled architecture does not contain a build infrastructure (devops-client, sourcecode-repo, build-pipeline, etc.), which might be the risk of a model missing critical assets (and thus not seeing their risks). If the architecture contains custom-developed parts, the pipeline where code gets developed and built needs to be part of the model.

Medium: **Missing Cloud Hardening**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

Cloud components should be hardened according to the cloud vendor best practices. This affects their configuration, auditing, and further areas.

Medium: **Missing Hardening**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* impact.

Technical assets with a Relative Attacker Attractiveness (RAA) value of 55 % or higher should be explicitly hardened taking best practices and vendor hardening guides into account.

Repudiation

n/a

Information Disclosure

Medium: **Missing Vault (Secret Storage)**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
In order to avoid the risk of secret leakage via config files (when attacked through vulnerabilities being able to read files like Path-Traversal and others), it is best practice to use a separate hardened process with proper authentication, authorization, and audit logging to access config secrets (like credentials, private keys, client certificates, etc.). This component is usually some kind of Vault.

Denial of Service

n/a

Elevation of Privilege

n/a

Assignment by Function

This chapter clusters and assigns the risks by functions which are most likely able to check and mitigate them: In total **9 potential risks** have been identified during the threat modeling process of which **0 should be checked by Business Side**, **3 should be checked by Architecture**, **3 should be checked by Development**, and **3 should be checked by Operations**.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Business Side

n/a

Architecture

Medium: **Missing Build Infrastructure**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
Include the build infrastructure in the model.

Medium: **Missing Identity Store**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
Include an identity store in the model if the application has a login.

Medium: **Missing Vault (Secret Storage)**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
Consider using a Vault (Secret Storage) to securely store and access config secrets (like credentials, private keys, client certificates, etc.).

Development

Elevated: **Cross-Site Scripting (XSS)**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Medium* impact.
Try to encode all values sent back to the browser and also handle DOM-manipulations in a safe way to avoid DOM-based XSS. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

Elevated: **SQL/NoSQL-Injection**: 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.
Try to use parameter binding to be safe from injection vulnerabilities. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

Operations

Medium: **Missing Cloud Hardening**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.
Apply hardening of all cloud components and services, taking special care to follow the individual risk descriptions (which depend on the cloud provider tags in the model).

Medium: **Missing Hardening**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* impact.
Try to apply all hardening best practices (like CIS benchmarks, OWASP recommendations, vendor recommendations, DevSec Hardening Framework, DBSAT for Oracle databases, and others).

RAA Analysis

For each technical asset the **"Relative Attacker Attractiveness"** (RAA) value was calculated in percent. The higher the RAA, the more interesting it is for an attacker to compromise the asset. The calculation algorithm takes the sensitivity ratings and quantities of stored and processed data into account as well as the communication links of the technical asset. Neighbouring assets to high-value RAA targets might receive an increase in their RAA value when they have a communication link towards that target ("Pivoting-Factor").

The following lists all technical assets sorted by their RAA value from highest (most attacker attractive) to lowest. This list can be used to prioritize on efforts relevant for the most attacker-attractive technical assets:

Technical asset paragraphs are clickable and link to the corresponding chapter.

database: RAA 100%
Gesicherte Benutzerdatenbank

app: RAA 58%
Gesicherte Backend Applikation

firewall1: RAA 40%
Firewall zwischen Client und WebApp

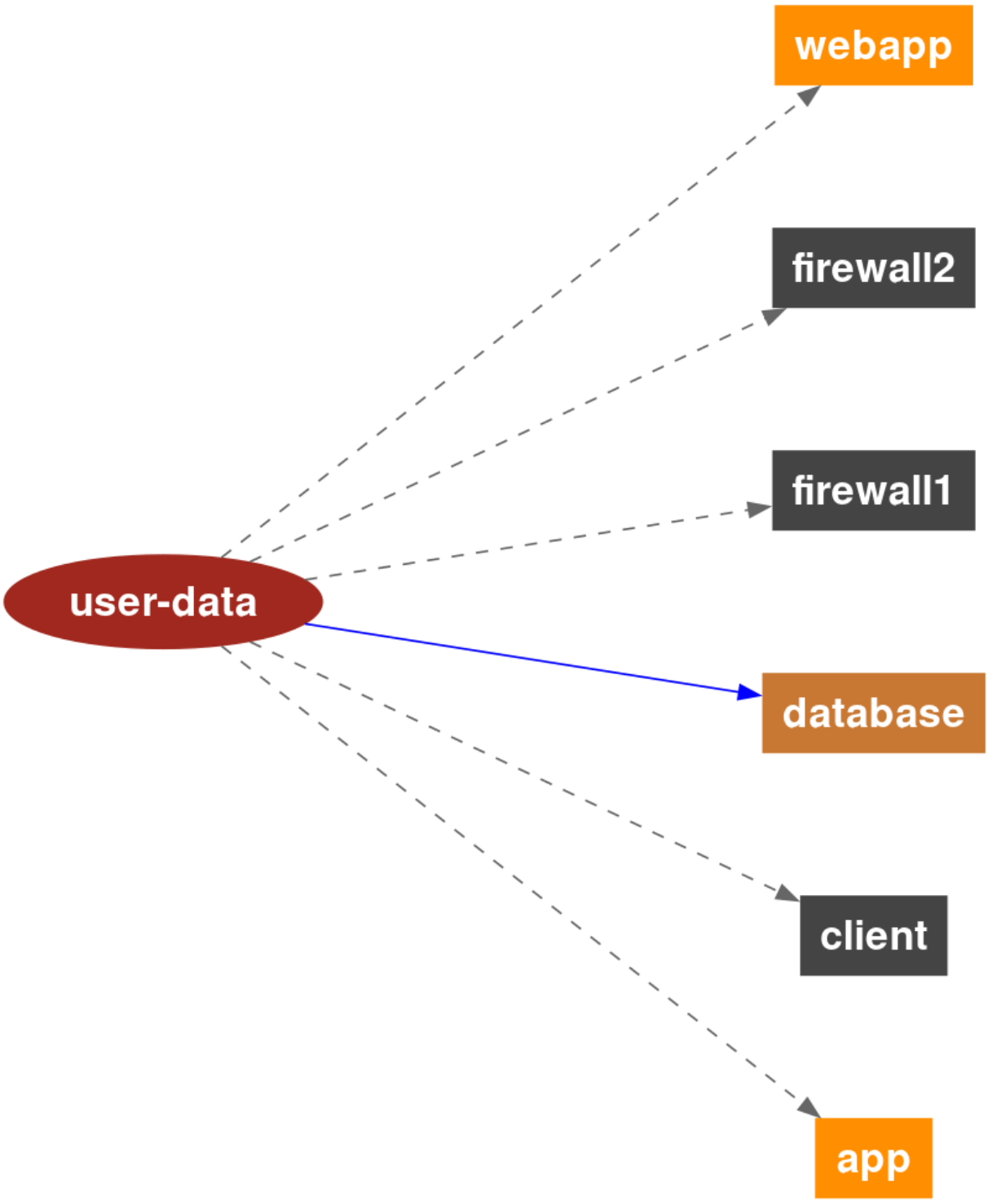
firewall2: RAA 40%
Firewall zwischen WebApp und App

client: RAA 12%
Externer Client

webapp: RAA 12%
Gesicherte WebApp in DMZ

Data Mapping

The following diagram was generated by Threagile based on the model input and gives a high-level distribution of data assets across technical assets. The color matches the identified data breach probability and risk level (see the "Data Breach Probabilities" chapter for more details). A solid line stands for *data is stored by the asset* and a dashed one means *data is processed by the asset*. For a full high-resolution version of this diagram please refer to the PNG image file alongside this report.



Out-of-Scope Assets: 0 Assets

This chapter lists all technical assets that have been defined as out-of-scope. Each one should be checked in the model whether it should better be included in the overall risk analysis:

Technical asset paragraphs are clickable and link to the corresponding chapter.

No technical assets have been defined as out-of-scope.

Potential Model Failures: 3 / 3 Risks

This chapter lists potential model failures where not all relevant assets have been modeled or the model might itself contain inconsistencies. Each potential model failure should be checked in the model against the architecture design:

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium: **Missing Build Infrastructure**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

The modeled architecture does not contain a build infrastructure (devops-client, sourcecode-repo, build-pipeline, etc.), which might be the risk of a model missing critical assets (and thus not seeing their risks). If the architecture contains custom-developed parts, the pipeline where code gets developed and built needs to be part of the model.

Medium: **Missing Identity Store**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

The modeled architecture does not contain an identity store, which might be the risk of a model missing critical assets (and thus not seeing their risks).

Medium: **Missing Vault (Secret Storage)**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

In order to avoid the risk of secret leakage via config files (when attacked through vulnerabilities being able to read files like Path-Traversal and others), it is best practice to use a separate hardened process with proper authentication, authorization, and audit logging to access config secrets (like credentials, private keys, client certificates, etc.). This component is usually some kind of Vault.

Questions: 0 / 0 Questions

This chapter lists custom questions that arose during the threat modeling process.

No custom questions arose during the threat modeling process.

Identified Risks by Vulnerability Category

In total **9 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **3 as elevated**, **6 as medium**, and **0 as low**.

These risks are distributed across **7 vulnerability categories**. The following sub-chapters of this section describe each identified risk category.

Cross-Site Scripting (XSS): 2 / 2 Risks

Description (Tampering): CWE 79

For each web application Cross-Site Scripting (XSS) risks might arise. In terms of the overall risk level take other applications running on the same domain into account as well.

Impact

If this risk remains unmitigated, attackers might be able to access individual victim sessions and steal or modify user data.

Detection Logic

In-scope web applications.

Risk Rating

The risk rating depends on the sensitivity of the data processed or stored in the web application.

False Positives

When the technical asset is not accessed via a browser-like component (i.e not by a human user initiating the request that gets passed through all components until it reaches the web application) this can be considered a false positive.

Mitigation (Development): XSS Prevention

Try to encode all values sent back to the browser and also handle DOM-manipulations in a safe way to avoid DOM-based XSS. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

ASVS Chapter: V5 - Validation, Sanitization and Encoding Verification Requirements
Cheat Sheet: Cross_Site_Scripting_Prevention_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **Cross-Site Scripting (XSS)** was found **2 times** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated Risk Severity

Cross-Site Scripting (XSS) risk at **app**: Exploitation likelihood is *Likely* with *Medium* impact.

cross-site-scripting@app

Unchecked

Cross-Site Scripting (XSS) risk at **webapp**: Exploitation likelihood is *Likely* with *Medium* impact.

cross-site-scripting@webapp

Unchecked

SQL/NoSQL-Injection: 1 / 1 Risk

Description (Tampering): CWE 89

When a database is accessed via database access protocols SQL/NoSQL-Injection risks might arise. The risk rating depends on the sensitivity technical asset itself and of the data assets processed or stored.

Impact

If this risk is unmitigated, attackers might be able to modify SQL/NoSQL queries to steal and modify data and eventually further escalate towards a deeper system penetration via code executions.

Detection Logic

Database accessed via typical database access protocols by in-scope clients.

Risk Rating

The risk rating depends on the sensitivity of the data stored inside the database.

False Positives

Database accesses by queries not consisting of parts controllable by the caller can be considered as false positives after individual review.

Mitigation (Development): SQL/NoSQL-Injection Prevention

Try to use parameter binding to be safe from injection vulnerabilities. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

ASVS Chapter: V5 - Validation, Sanitization and Encoding Verification Requirements
Cheat Sheet: SQL_Injection_Prevention_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **SQL/NoSQL-Injection** was found **1 time** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated Risk Severity

SQL/NoSQL-Injection risk at **app** against database **database** via **to-database**: Exploitation likelihood is *Very Likely* with *Medium* impact.

sql-nosql-injection@app@database@app>to-database

Unchecked

Missing Build Infrastructure: 1 / 1 Risk

Description (Tampering): CWE 1127

The modeled architecture does not contain a build infrastructure (devops-client, sourcecode-repo, build-pipeline, etc.), which might be the risk of a model missing critical assets (and thus not seeing their risks). If the architecture contains custom-developed parts, the pipeline where code gets developed and built needs to be part of the model.

Impact

If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model due to critical build infrastructure components missing in the model.

Detection Logic

Models with in-scope custom-developed parts missing in-scope development (code creation) and build infrastructure components (devops-client, sourcecode-repo, build-pipeline, etc.).

Risk Rating

The risk rating depends on the highest sensitivity of the in-scope assets running custom-developed parts.

False Positives

Models not having any custom-developed parts can be considered as false positives after individual review.

Mitigation (Architecture): Build Pipeline Hardening

Include the build infrastructure in the model.

ASVS Chapter: V1 - Architecture, Design and Threat Modeling Requirements
Cheat Sheet: Attack_Surface_Analysis_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **Missing Build Infrastructure** was found **1 time** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Build Infrastructure in the threat model (referencing asset **app** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-build-infrastructure@app

Unchecked

Missing Cloud Hardening: 1 / 1 Risk

Description (Tampering): CWE 1008

Cloud components should be hardened according to the cloud vendor best practices. This affects their configuration, auditing, and further areas.

Impact

If this risk is unmitigated, attackers might access cloud components in an unintended way.

Detection Logic

In-scope cloud components (either residing in cloud trust boundaries or more specifically tagged with cloud provider types).

Risk Rating

The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

False Positives

Cloud components not running parts of the target architecture can be considered as false positives after individual review.

Mitigation (Operations): Cloud Hardening

Apply hardening of all cloud components and services, taking special care to follow the individual risk descriptions (which depend on the cloud provider tags in the model).

For **Amazon Web Services (AWS)**: Follow the *CIS Benchmark for Amazon Web Services* (see also the automated checks of cloud audit tools like *"PacBot", "CloudSploit", "CloudMapper", "ScoutSuite", or "Prowler AWS CIS Benchmark Tool"*).
For EC2 and other servers running Amazon Linux, follow the *CIS Benchmark for Amazon Linux* and switch to IMDSv2.
For S3 buckets follow the *Security Best Practices for Amazon S3* at https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html to avoid accidental leakage.
Also take a look at some of these tools: https://github.com/toniblyx/my-arsenal-of-aws-security-tools

For **Microsoft Azure**: Follow the *CIS Benchmark for Microsoft Azure* (see also the automated checks of cloud audit tools like *"CloudSploit" or "ScoutSuite"*).

For **Google Cloud Platform**: Follow the *CIS Benchmark for Google Cloud Computing Platform* (see also the automated checks of cloud audit tools like *"CloudSploit" or "ScoutSuite"*).

For **Oracle Cloud Platform**: Follow the hardening best practices (see also the automated checks of cloud audit tools like *"CloudSploit"*).

ASVS Chapter: V1 - Architecture, Design and Threat Modeling Requirements
Cheat Sheet: Attack_Surface_Analysis_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **Missing Cloud Hardening** was found **1 time** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Cloud Hardening risk at **database**: Exploitation likelihood is *Unlikely* with *High* impact.

missing-cloud-hardening@database

Unchecked

Missing Hardening: 2 / 2 Risks

Description (Tampering): CWE 16

Technical assets with a Relative Attacker Attractiveness (RAA) value of 55 % or higher should be explicitly hardened taking best practices and vendor hardening guides into account.

Impact

If this risk remains unmitigated, attackers might be able to easier attack high-value targets.

Detection Logic

In-scope technical assets with RAA values of 55 % or higher. Generally for high-value targets like datastores, application servers, identity providers and ERP systems this limit is reduced to 40 %

Risk Rating

The risk rating depends on the sensitivity of the data processed or stored in the technical asset.

False Positives

Usually no false positives.

Mitigation (Operations): System Hardening

Try to apply all hardening best practices (like CIS benchmarks, OWASP recommendations, vendor recommendations, DevSec Hardening Framework, DBSAT for Oracle databases, and others).

ASVS Chapter: V14 - Configuration Verification Requirements
Cheat Sheet: Attack_Surface_Analysis_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **Missing Hardening** was found **2 times** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Hardening risk at **app**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@app

Unchecked

Missing Hardening risk at **database**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@database

Unchecked

Missing Identity Store: 1 / 1 Risk

Description (Spoofing): CWE 287

The modeled architecture does not contain an identity store, which might be the risk of a model missing critical assets (and thus not seeing their risks).

Impact

If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model in the identity provider/store that is currently missing in the model.

Detection Logic

Models with authenticated data-flows authorized via enduser-identity missing an in-scope identity store.

Risk Rating

The risk rating depends on the sensitivity of the enduser-identity authorized technical assets and their data assets processed and stored.

False Positives

Models only offering data/services without any real authentication need can be considered as false positives after individual review.

Mitigation (Architecture): Identity Store

Include an identity store in the model if the application has a login.

ASVS Chapter: V2 - Authentication Verification Requirements
Cheat Sheet: Authentication_Cheat_Sheet

Check

Are recommendations from the linked cheat sheet and referenced ASVS chapter applied?

Risk Findings

The risk **Missing Identity Store** was found **1 time** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Identity Store in the threat model (referencing asset **webapp** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-identity-store@webapp

Unchecked

Missing Vault (Secret Storage): 1 / 1 Risk

Description (Information Disclosure): CWE 522

In order to avoid the risk of secret leakage via config files (when attacked through vulnerabilities being able to read files like Path-Traversal and others), it is best practice to use a separate hardened process with proper authentication, authorization, and audit logging to access config secrets (like credentials, private keys, client certificates, etc.). This component is usually some kind of Vault.

Impact

If this risk is unmitigated, attackers might be able to easier steal config secrets (like credentials, private keys, client certificates, etc.) once a vulnerability to access files is present and exploited.

Detection Logic

Models without a Vault (Secret Storage).

Risk Rating

The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

False Positives

Models where no technical assets have any kind of sensitive config data to protect can be considered as false positives after individual review.

Mitigation (Architecture): Vault (Secret Storage)

Consider using a Vault (Secret Storage) to securely store and access config secrets (like credentials, private keys, client certificates, etc.).

ASVS Chapter: V6 - Stored Cryptography Verification Requirements
Cheat Sheet: Cryptographic_Storage_Cheat_Sheet

Check

Is a Vault (Secret Storage) in place?

Risk Findings

The risk **Missing Vault (Secret Storage)** was found **1 time** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Vault (Secret Storage) in the threat model (referencing asset **app** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-vault@app

Unchecked

Identified Risks by Technical Asset

In total **9 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **3 as elevated**, **6 as medium**, and **0 as low**.

These risks are distributed across **6 in-scope technical assets**. The following sub-chapters of this section describe each identified risk grouped by technical asset. The RAA value of a technical asset is the calculated "Relative Attacker Attractiveness" value in percent.

app: 5 / 5 Risks

Description

Gesicherte Backend Applikation

Identified Risks of Asset

Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated Risk Severity

SQL/NoSQL-Injection risk at **app** against database **database** via **to-database**: Exploitation likelihood is *Very Likely* with *Medium* impact.

sql-nosql-injection@app@database@app>to-database

Unchecked

Cross-Site Scripting (XSS) risk at **app**: Exploitation likelihood is *Likely* with *Medium* impact.

cross-site-scripting@app

Unchecked

Medium Risk Severity

Missing Build Infrastructure in the threat model (referencing asset **app** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-build-infrastructure@app

Unchecked

Missing Vault (Secret Storage) in the threat model (referencing asset **app** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-vault@app

Unchecked

Missing Hardening risk at **app**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@app

Unchecked

Asset Information

ID:	app
Type:	process
Usage:	business
RAA:	58 %
Size:	service

Technology:	application-server
Tags:	none
Internet:	false
Machine:	virtual
Encryption:	data-with-symmetric-shared-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	true
Client by Human:	false
Data Processed:	user-data
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:		

Outgoing Communication Links: 1

Target technical asset names are clickable and link to the corresponding chapter.

to-database (outgoing)

Verschlüsselte Verbindung zur DB

Target:	database
Protocol:	jdbc-encrypted
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Sent:	user-data

Data Received: user-data

Incoming Communication Links: 1

Source technical asset names are clickable and link to the corresponding chapter.

to-app (incoming)

Übergabe an Backend-App

Source:	firewall2
Protocol:	ssh-tunnel
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Received:	user-data
Data Sent:	user-data

webapp: 2 / 2 Risks

Description

Gesicherte WebApp in DMZ

Identified Risks of Asset

Risk finding paragraphs are clickable and link to the corresponding chapter.

Elevated Risk Severity

Cross-Site Scripting (XSS) risk at **webapp**: Exploitation likelihood is *Likely* with *Medium* impact.

cross-site-scripting@webapp

Unchecked

Medium Risk Severity

Missing Identity Store in the threat model (referencing asset **webapp** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-identity-store@webapp

Unchecked

Asset Information

ID:	webapp
Type:	process
Usage:	business
RAA:	12 %
Size:	application
Technology:	web-server
Tags:	none
Internet:	true
Machine:	virtual
Encryption:	data-with-symmetric-shared-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	true
Client by Human:	false
Data Processed:	user-data
Data Stored:	none

Formats Accepted: JSON

Asset Rating

Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:		

Outgoing Communication Links: 1

Target technical asset names are clickable and link to the corresponding chapter.

to-firewall2 (outgoing)

Verbindungs zur firewall2

Target:	firewall2
Protocol:	ssh-tunnel
Encrypted:	true
Authentication:	client-certificate
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Sent:	user-data
Data Received:	none

Incoming Communication Links: 1

Source technical asset names are clickable and link to the corresponding chapter.

to-webapp (incoming)

Übergabe an WebApp

Source:	firewall1
Protocol:	ssh-tunnel
Encrypted:	true

Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Received:	user-data
Data Sent:	user-data

database: 1 / 1 Risk

Description

Gesicherte Benutzerdatenbank

Identified Risks of Asset

Risk finding paragraphs are clickable and link to the corresponding chapter.

Medium Risk Severity

Missing Hardening risk at **database**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@database

Unchecked

Asset Information

ID:	database
Type:	datastore
Usage:	business
RAA:	100 %
Size:	service
Technology:	database
Tags:	none
Internet:	false
Machine:	virtual
Encryption:	data-with-symmetric-shared-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	false
Client by Human:	false
Data Processed:	none
Data Stored:	user-data
Formats Accepted:	JSON

Asset Rating

Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)

Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:		

Incoming Communication Links: 1

Source technical asset names are clickable and link to the corresponding chapter.

to-database (incoming)

Verschlüsselte Verbindung zur DB

Source:	app
Protocol:	jdbc-encrypted
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Received:	user-data
Data Sent:	user-data

client: 0 / 0 Risks

Description

Externer Client

Identified Risks of Asset

No risks were identified.

Asset Information

ID:	client
Type:	external-entity
Usage:	business
RAA:	12 %
Size:	component
Technology:	client-system
Tags:	none
Internet:	false
Machine:	physical
Encryption:	data-with-enduser-individual-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	false
Client by Human:	true
Data Processed:	user-data
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	users	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:		

Outgoing Communication Links: 1

Target technical asset names are clickable and link to the corresponding chapter.

to-firewall1 (outgoing)

HTTPS-Verbindung mit Authentifizierung

Target:	firewall1
Protocol:	https
Encrypted:	true
Authentication:	two-factor
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Sent:	user-data
Data Received:	none

firewall1: 0 / 0 Risks

Description

Firewall zwischen Client und WebApp

Identified Risks of Asset

No risks were identified.

Asset Information

ID:	firewall1
Type:	process
Usage:	business
RAA:	40 %
Size:	component
Technology:	gateway
Tags:	none
Internet:	false
Machine:	virtual
Encryption:	data-with-enduser-individual-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	false
Client by Human:	false
Data Processed:	user-data
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	Schutz vor unautorisierten Zugriffen vom Client zur WebApp	

Outgoing Communication Links: 1

Target technical asset names are clickable and link to the corresponding chapter.

to-webapp (outgoing)

Übergabe an WebApp

Target:	webapp
Protocol:	ssh-tunnel
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Sent:	user-data
Data Received:	user-data

Incoming Communication Links: 1

Source technical asset names are clickable and link to the corresponding chapter.

to-firewall1 (incoming)

HTTPS-Verbindung mit Authentifizierung

Source:	client
Protocol:	https
Encrypted:	true
Authentication:	two-factor
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Received:	user-data
Data Sent:	none

firewall2: 0 / 0 Risks

Description

Firewall zwischen WebApp und App

Identified Risks of Asset

No risks were identified.

Asset Information

ID:	firewall2
Type:	process
Usage:	business
RAA:	40 %
Size:	component
Technology:	gateway
Tags:	none
Internet:	false
Machine:	virtual
Encryption:	data-with-enduser-individual-key
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	false
Client by Human:	false
Data Processed:	user-data
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	Schutz vor direkten Zugriffen von WebApp auf das interne Backend	

Outgoing Communication Links: 1

Target technical asset names are clickable and link to the corresponding chapter.

to-app (outgoing)

Übergabe an Backend-App

Target:	app
Protocol:	ssh-tunnel
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	true
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Sent:	user-data
Data Received:	user-data

Incoming Communication Links: 1

Source technical asset names are clickable and link to the corresponding chapter.

to-firewall2 (incoming)

Verbindungs zur firewall2

Source:	webapp
Protocol:	ssh-tunnel
Encrypted:	true
Authentication:	client-certificate
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	true
Data Received:	user-data
Data Sent:	none

Identified Data Breach Probabilities by Data Asset

In total **9 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **3 as elevated**, **6 as medium**, and **0 as low**.

These risks are distributed across **1 data assets**. The following sub-chapters of this section describe the derived data breach probabilities grouped by data asset.

Technical asset names and risk IDs are clickable and link to the corresponding chapter.

user-data: 6 / 6 Risks

Benutzerdaten mit Authentifizierungsinformationen

ID:	user-data	
Usage:	business	
Quantity:	many	
Tags:	pii	
Origin:	users	
Owner:	IT	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	Benutzerdaten erfordern Schutz und sind nun abgesichert	
Processed by:	app, client, firewall1, firewall2, webapp	
Stored by:	database	
Sent via:	to-webapp, to-firewall2, to-firewall1, to-database, to-app	
Received via:	to-webapp, to-database, to-app	
Data Breach:	**probable**	
Data Breach Risks:	This data asset has data breach potential because of 6 remaining risks:	

Probable: missing-cloud-hardening@database

Probable: sql-nosql-injection@app@database@app>to-database

Possible: cross-site-scripting@app

Possible: cross-site-scripting@webapp

Improbable: missing-hardening@app

Improbable: missing-hardening@database

Trust Boundaries

In total **2 trust boundaries** have been modeled during the threat modeling process.

database
Netzwerksegmentierung

ID:	database
Type:	network-cloud-provider
Tags:	none
Assets inside:	app, database
Boundaries nested:	none

dmz
Demilitarisierte Zone

ID:	dmz
Type:	network-on-prem
Tags:	none
Assets inside:	webapp
Boundaries nested:	none

Shared Runtimes

In total **0 shared runtime** has been modeled during the threat modeling process.

Risk Rules Checked by Threagile

Threagile Version: 1.0.0
Threagile Build Timestamp: 20240730113903
Threagile Execution Timestamp: 20250710002342
Model Filename: /app/work/threagile_secure.yaml
Model Hash (SHA256): 97176974faaec679a01e67249c4ffdd9b2fa40eb386c2ca981edc56c2eed8f3a

Threagile (see https://threagile.io for more details) is an open-source toolkit for agile threat modeling, created by Christian Schneider (https://christian-schneider.net): It allows to model an architecture with its assets in an agile fashion as a YAML file directly inside the IDE. Upon execution of the Threagile toolkit all standard risk rules (as well as individual custom rules if present) are checked against the architecture model. At the time the Threagile toolkit was executed on the model input file the following risk rules were checked:

Accidental Secret Leak
accidental-secret-leak

STRIDE:	Information Disclosure
Description:	Sourcecode repositories (including their histories) as well as artifact registries can accidentally contain secrets like checked-in or packaged-in passwords, API tokens, certificates, crypto keys, etc.
Detection:	In-scope sourcecode repositories and artifact registries.
Rating:	The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Code Backdooring
code-backdooring

STRIDE:	Tampering
Description:	For each build-pipeline component Code Backdooring risks might arise where attackers compromise the build-pipeline in order to let backdoored artifacts be shipped into production. Aside from direct code backdooring this includes backdooring of dependencies and even of more lower-level build infrastructure, like backdooring compilers (similar to what the XcodeGhost malware did) or dependencies.
Detection:	In-scope development relevant technical assets which are either accessed by out-of-scope unmanaged developer clients and/or are directly accessed by any kind of internet-located (non-VPN) component or are themselves directly located on the internet.
Rating:	The risk rating depends on the confidentiality and integrity rating of the code being handled and deployed as well as the placement/calling of this technical asset on/from the internet.

Container Base Image Backdooring
container-baseimage-backdooring

STRIDE: Tampering
Description: When a technical asset is built using container technologies, Base Image Backdooring risks might arise where base images and other layers used contain vulnerable components or backdoors.
Detection: In-scope technical assets running as containers.
Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets.

Container Platform Escape
container-platform-escape

STRIDE: Elevation of Privilege
Description: Container platforms are especially interesting targets for attackers as they host big parts of a containerized runtime infrastructure. When not configured and operated with security best practices in mind, attackers might exploit a vulnerability inside an container and escape towards the platform as highly privileged users. These scenarios might give attackers capabilities to attack every other container as owning the container platform (via container escape attacks) equals to owning every container.
Detection: In-scope container platforms.
Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Cross-Site Request Forgery (CSRF)
cross-site-request-forgery

STRIDE: Spoofing
Description: When a web application is accessed via web protocols Cross-Site Request Forgery (CSRF) risks might arise.
Detection: In-scope web applications accessed via typical web access protocols.
Rating: The risk rating depends on the integrity rating of the data sent across the communication link.

Cross-Site Scripting (XSS)
cross-site-scripting

STRIDE: Tampering
Description: For each web application Cross-Site Scripting (XSS) risks might arise. In terms of the overall risk level take other applications running on the same domain into account as well.
Detection: In-scope web applications.
Rating: The risk rating depends on the sensitivity of the data processed or stored in the web application.

DoS-risky Access Across Trust-Boundary

dos-risky-access-across-trust-boundary

STRIDE: Denial of Service

Description: Assets accessed across trust boundaries with critical or mission-critical availability rating are more prone to Denial-of-Service (DoS) risks.

Detection: In-scope technical assets (excluding load-balancer) with availability rating of critical or higher which have incoming data-flows across a network trust-boundary (excluding devops usage).

Rating: Matching technical assets with availability rating of critical or higher are at low risk. When the availability rating is mission-critical and neither a VPN nor IP filter for the incoming data-flow nor redundancy for the asset is applied, the risk-rating is considered medium.

Incomplete Model

incomplete-model

STRIDE: Information Disclosure

Description: When the threat model contains unknown technologies or transfers data over unknown protocols, this is an indicator for an incomplete model.

Detection: All technical assets and communication links with technology type or protocol type specified as unknown.

Rating: low

LDAP-Injection

ldap-injection

STRIDE: Tampering

Description: When an LDAP server is accessed LDAP-Injection risks might arise. The risk rating depends on the sensitivity of the LDAP server itself and of the data assets processed or stored.

Detection: In-scope clients accessing LDAP servers via typical LDAP access protocols.

Rating: The risk rating depends on the sensitivity of the LDAP server itself and of the data assets processed or stored.

Missing Authentication

missing-authentication

STRIDE: Elevation of Privilege

Description: Technical assets (especially multi-tenant systems) should authenticate incoming requests when the asset processes or stores sensitive data.

Detection: In-scope technical assets (except load-balancer, reverse-proxy, service-registry, waf, ids, and ips and in-process calls) should authenticate incoming requests when the asset processes or stores sensitive data. This is especially the case for all multi-tenant assets (there even non-sensitive ones).

Rating: The risk rating (medium or high) depends on the sensitivity of the data sent across

the communication link. Monitoring callers are exempted from this risk.

Missing Two-Factor Authentication (2FA)

missing-authentication-second-factor

STRIDE: Elevation of Privilege

Description: Technical assets (especially multi-tenant systems) should authenticate incoming requests with two-factor (2FA) authentication when the asset processes or stores highly sensitive data (in terms of confidentiality, integrity, and availability) and is accessed by humans.

Detection: In-scope technical assets (except load-balancer, reverse-proxy, waf, ids, and ips) should authenticate incoming requests via two-factor authentication (2FA) when the asset processes or stores highly sensitive data (in terms of confidentiality, integrity, and availability) and is accessed by a client used by a human user.

Rating: medium

Missing Build Infrastructure

missing-build-infrastructure

STRIDE: Tampering

Description: The modeled architecture does not contain a build infrastructure (devops-client, sourcecode-repo, build-pipeline, etc.), which might be the risk of a model missing critical assets (and thus not seeing their risks). If the architecture contains custom-developed parts, the pipeline where code gets developed and built needs to be part of the model.

Detection: Models with in-scope custom-developed parts missing in-scope development (code creation) and build infrastructure components (devops-client, sourcecode-repo, build-pipeline, etc.).

Rating: The risk rating depends on the highest sensitivity of the in-scope assets running custom-developed parts.

Missing Cloud Hardening

missing-cloud-hardening

STRIDE: Tampering

Description: Cloud components should be hardened according to the cloud vendor best practices. This affects their configuration, auditing, and further areas.

Detection: In-scope cloud components (either residing in cloud trust boundaries or more specifically tagged with cloud provider types).

Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Missing File Validation

missing-file-validation

STRIDE: Spoofing

Description: When a technical asset accepts files, these input files should be strictly validated about filename and type.
Detection: In-scope technical assets with custom-developed code accepting file data formats.
Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Missing Hardening
missing-hardening

STRIDE: Tampering
Description: Technical assets with a Relative Attacker Attractiveness (RAA) value of 55 % or higher should be explicitly hardened taking best practices and vendor hardening guides into account.
Detection: In-scope technical assets with RAA values of 55 % or higher. Generally for high-value targets like datastores, application servers, identity providers and ERP systems this limit is reduced to 40 %
Rating: The risk rating depends on the sensitivity of the data processed or stored in the technical asset.

Missing Identity Propagation
missing-identity-propagation

STRIDE: Elevation of Privilege
Description: Technical assets (especially multi-tenant systems), which usually process data for endusers should authorize every request based on the identity of the enduser when the data flow is authenticated (i.e. non-public). For DevOps usages at least a technical-user authorization is required.
Detection: In-scope service-like technical assets which usually process data based on enduser requests, if authenticated (i.e. non-public), should authorize incoming requests based on the propagated enduser identity when their rating is sensitive. This is especially the case for all multi-tenant assets (there even less-sensitive rated ones). DevOps usages are exempted from this risk.
Rating: The risk rating (medium or high) depends on the confidentiality, integrity, and availability rating of the technical asset.

Missing Identity Provider Isolation
missing-identity-provider-isolation

STRIDE: Elevation of Privilege
Description: Highly sensitive identity provider assets and their identity datastores should be isolated from other assets by their own network segmentation trust-boundary (execution-environment boundaries do not count as network isolation).
Detection: In-scope identity provider assets and their identity datastores when surrounded by other (not identity-related) assets (without a network trust-boundary in-between).

This risk is especially prevalent when other non-identity related assets are within the same execution environment (i.e. same database or same application server).

Rating: Default is high impact. The impact is increased to very-high when the asset missing the trust-boundary protection is rated as strictly-confidential or mission-critical.

Missing Identity Store

missing-identity-store

STRIDE: Spoofing

Description: The modeled architecture does not contain an identity store, which might be the risk of a model missing critical assets (and thus not seeing their risks).

Detection: Models with authenticated data-flows authorized via enduser-identity missing an in-scope identity store.

Rating: The risk rating depends on the sensitivity of the enduser-identity authorized technical assets and their data assets processed and stored.

Missing Network Segmentation

missing-network-segmentation

STRIDE: Elevation of Privilege

Description: Highly sensitive assets and/or datastores residing in the same network segment than other lower sensitive assets (like webservers or content management systems etc.) should be better protected by a network segmentation trust-boundary.

Detection: In-scope technical assets with high sensitivity and RAA values as well as datastores when surrounded by assets (without a network trust-boundary in-between) which are of type client-system, web-server, web-application, cms, web-service-rest, web-service-soap, build-pipeline, sourcecode-repository, monitoring, or similar and there is no direct connection between these (hence no requirement to be so close to each other).

Rating: Default is low risk. The risk is increased to medium when the asset missing the trust-boundary protection is rated as strictly-confidential or mission-critical.

Missing Vault (Secret Storage)

missing-vault

STRIDE: Information Disclosure

Description: In order to avoid the risk of secret leakage via config files (when attacked through vulnerabilities being able to read files like Path-Traversal and others), it is best practice to use a separate hardened process with proper authentication, authorization, and audit logging to access config secrets (like credentials, private keys, client certificates, etc.). This component is usually some kind of Vault.

Detection: Models without a Vault (Secret Storage).

Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Missing Vault Isolation
missing-vault-isolation

STRIDE: Elevation of Privilege

Description: Highly sensitive vault assets and their datastores should be isolated from other assets by their own network segmentation trust-boundary (execution-environment boundaries do not count as network isolation).

Detection: In-scope vault assets when surrounded by other (not vault-related) assets (without a network trust-boundary in-between). This risk is especially prevalent when other non-vault related assets are within the same execution environment (i.e. same database or same application server).

Rating: Default is medium impact. The impact is increased to high when the asset missing the trust-boundary protection is rated as strictly-confidential or mission-critical.

Missing Web Application Firewall (WAF)
missing-waf

STRIDE: Tampering

Description: To have a first line of filtering defense, security architectures with web-services or web-applications should include a WAF in front of them. Even though a WAF is not a replacement for security (all components must be secure even without a WAF) it adds another layer of defense to the overall system by delaying some attacks and having easier attack alerting through it.

Detection: In-scope web-services and/or web-applications accessed across a network trust boundary not having a Web Application Firewall (WAF) in front of them.

Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Mixed Targets on Shared Runtime
mixed-targets-on-shared-runtime

STRIDE: Elevation of Privilege

Description: Different attacker targets (like frontend and backend/datastore components) should not be running on the same shared (underlying) runtime.

Detection: Shared runtime running technical assets of different trust-boundaries is at risk. Also mixing backend/datastore with frontend components on the same shared runtime is considered a risk.

Rating: The risk rating (low or medium) depends on the confidentiality, integrity, and availability rating of the technical asset running on the shared runtime.

Path-Traversal
path-traversal

STRIDE: Information Disclosure

Description: When a filesystem is accessed Path-Traversal or Local-File-Inclusion (LFI) risks might arise. The risk rating depends on the sensitivity of the technical asset itself

and of the data assets processed or stored.

Detection: Filesystems accessed by in-scope callers.

Rating: The risk rating depends on the sensitivity of the data stored inside the technical asset.

Push instead of Pull Deployment

push-instead-of-pull-deployment

STRIDE: Tampering

Description: When comparing push-based vs. pull-based deployments from a security perspective, pull-based deployments improve the overall security of the deployment targets. Every exposed interface of a production system to accept a deployment increases the attack surface of the production system, thus a pull-based approach exposes less attack surface relevant interfaces.

Detection: Models with build pipeline components accessing in-scope targets of deployment (in a non-readonly way) which are not build-related components themselves.

Rating: The risk rating depends on the highest sensitivity of the deployment targets running custom-developed parts.

Search-Query Injection

search-query-injection

STRIDE: Tampering

Description: When a search engine server is accessed Search-Query Injection risks might arise.

Detection: In-scope clients accessing search engine servers via typical search access protocols.

Rating: The risk rating depends on the sensitivity of the search engine server itself and of the data assets processed or stored.

Server-Side Request Forgery (SSRF)

server-side-request-forgery

STRIDE: Information Disclosure

Description: When a server system (i.e. not a client) is accessing other server systems via typical web protocols Server-Side Request Forgery (SSRF) or Local-File-Inclusion (LFI) or Remote-File-Inclusion (RFI) risks might arise.

Detection: In-scope non-client systems accessing (using outgoing communication links) targets with either HTTP or HTTPS protocol.

Rating: The risk rating (low or medium) depends on the sensitivity of the data assets receivable via web protocols from targets within the same network trust-boundary as well on the sensitivity of the data assets receivable via web protocols from the target asset itself. Also for cloud-based environments the exploitation impact is at least medium, as cloud backend services can be attacked via SSRF.

Service Registry Poisoning

service-registry-poisoning

STRIDE: Spoofing

Description: When a service registry used for discovery of trusted service endpoints Service Registry Poisoning risks might arise.

Detection: In-scope service registries.

Rating: The risk rating depends on the sensitivity of the technical assets accessing the service registry as well as the data assets processed or stored.

SQL/NoSQL-Injection

sql-nosql-injection

STRIDE: Tampering

Description: When a database is accessed via database access protocols SQL/NoSQL-Injection risks might arise. The risk rating depends on the sensitivity technical asset itself and of the data assets processed or stored.

Detection: Database accessed via typical database access protocols by in-scope clients.

Rating: The risk rating depends on the sensitivity of the data stored inside the database.

Unchecked Deployment

unchecked-deployment

STRIDE: Tampering

Description: For each build-pipeline component Unchecked Deployment risks might arise when the build-pipeline does not include established DevSecOps best-practices. DevSecOps best-practices scan as part of CI/CD pipelines for vulnerabilities in source- or byte-code, dependencies, container layers, and dynamically against running test systems. There are several open-source and commercial tools existing in the categories DAST, SAST, and IAST.

Detection: All development-relevant technical assets.

Rating: The risk rating depends on the highest rating of the technical assets and data assets processed by deployment-receiving targets.

Unencrypted Technical Assets

unencrypted-asset

STRIDE: Information Disclosure

Description: Due to the confidentiality rating of the technical asset itself and/or the processed data assets this technical asset must be encrypted. The risk rating depends on the sensitivity technical asset itself and of the data assets stored.

Detection: In-scope unencrypted technical assets (excluding reverse-proxy, load-balancer, waf, ids, ips and embedded components like library) storing data assets rated at least as confidential or critical. For technical assets storing data assets rated as strictly-confidential or mission-critical the encryption must be of type data-with-enduser-individual-key.

Rating: Depending on the confidentiality rating of the stored data-assets either medium or high risk.

Unencrypted Communication
unencrypted-communication

STRIDE: Information Disclosure

Description: Due to the confidentiality and/or integrity rating of the data assets transferred over the communication link this connection must be encrypted.

Detection: Unencrypted technical communication links of in-scope technical assets (excluding monitoring traffic as well as local-file-access and in-process-library-call) transferring sensitive data.

Rating: Depending on the confidentiality rating of the transferred data-assets either medium or high risk.

Unguarded Access From Internet
unguarded-access-from-internet

STRIDE: Elevation of Privilege

Description: Internet-exposed assets must be guarded by a protecting service, application, or reverse-proxy.

Detection: In-scope technical assets (excluding load-balancer) with confidentiality rating of confidential (or higher) or with integrity rating of critical (or higher) when accessed directly from the internet. All web-server, web-application, reverse-proxy, waf, and gateway assets are exempted from this risk when they do not consist of custom developed code and the data-flow only consists of HTTP or FTP protocols. Access from monitoring systems as well as VPN-protected connections are exempted.

Rating: The matching technical assets are at low risk. When either the confidentiality rating is strictly-confidential or the integrity rating is mission-critical, the risk-rating is considered medium. For assets with RAA values higher than 40 % the risk-rating increases.

Unguarded Direct Datastore Access
unguarded-direct-datastore-access

STRIDE: Elevation of Privilege

Description: Datastores accessed across trust boundaries must be guarded by some protecting service or application.

Detection: In-scope technical assets of type datastore (except identity-store-ldap when accessed from identity-provider and file-server when accessed via file transfer protocols) with confidentiality rating of confidential (or higher) or with integrity rating of critical (or higher) which have incoming data-flows from assets outside across a network trust-boundary. DevOps config and deployment access is excluded from this risk.

Rating: The matching technical assets are at low risk. When either the confidentiality rating is strictly-confidential or the integrity rating is mission-critical, the risk-rating is considered medium. For assets with RAA values higher than 40 % the risk-rating increases.

Unnecessary Communication Link
unnecessary-communication-link

STRIDE: Elevation of Privilege

Description: When a technical communication link does not send or receive any data assets, this is an indicator for an unnecessary communication link (or for an incomplete model).

Detection: In-scope technical assets' technical communication links not sending or receiving any data assets.

Rating: low

Unnecessary Data Asset
unnecessary-data-asset

STRIDE: Elevation of Privilege

Description: When a data asset is not processed or stored by any data assets and also not transferred by any communication links, this is an indicator for an unnecessary data asset (or for an incomplete model).

Detection: Modelled data assets not processed or stored by any data assets and also not transferred by any communication links.

Rating: low

Unnecessary Data Transfer
unnecessary-data-transfer

STRIDE: Elevation of Privilege

Description: When a technical asset sends or receives data assets, which it neither processes or stores this is an indicator for unnecessarily transferred data (or for an incomplete model). When the unnecessarily transferred data assets are sensitive, this poses an unnecessary risk of an increased attack surface.

Detection: In-scope technical assets sending or receiving sensitive data assets which are neither processed nor stored by the technical asset are flagged with this risk. The risk rating (low or medium) depends on the confidentiality, integrity, and availability rating of the technical asset. Monitoring data is exempted from this risk.

Rating: The risk assessment is depending on the confidentiality and integrity rating of the transferred data asset either low or medium.

Unnecessary Technical Asset
unnecessary-technical-asset

STRIDE: Elevation of Privilege

Description: When a technical asset does not process or store any data assets, this is an

indicator for an unnecessary technical asset (or for an incomplete model). This is also the case if the asset has no communication links (either outgoing or incoming).

Detection: Technical assets not processing or storing any data assets.

Rating: low

Untrusted Deserialization

untrusted-deserialization

STRIDE: Tampering

Description: When a technical asset accepts data in a specific serialized form (like Java or .NET serialization), Untrusted Deserialization risks might arise.

Detection: In-scope technical assets accepting serialization data formats (including EJB and RMI protocols).

Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data assets processed and stored.

Wrong Communication Link Content

wrong-communication-link-content

STRIDE: Information Disclosure

Description: When a communication link is defined as readonly, but does not receive any data asset, or when it is defined as not readonly, but does not send any data asset, it is likely to be a model failure.

Detection: Communication links with inconsistent data assets being sent/received not matching their readonly flag or otherwise inconsistent protocols not matching the target technology type.

Rating: low

Wrong Trust Boundary Content

wrong-trust-boundary-content

STRIDE: Elevation of Privilege

Description: When a trust boundary of type network-policy-namespace-isolation contains non-container assets it is likely to be a model failure.

Detection: Trust boundaries which should only contain containers, but have different assets inside.

Rating: low

XML External Entity (XXE)

xml-external-entity

STRIDE: Information Disclosure

Description: When a technical asset accepts data in XML format, XML External Entity (XXE) risks might arise.

Detection: In-scope technical assets accepting XML data formats.

Rating: The risk rating depends on the sensitivity of the technical asset itself and of the data

assets processed and stored. Also for cloud-based environments the exploitation impact is at least medium, as cloud backend services can be attacked via SSRF (and XXE vulnerabilities are often also SSRF vulnerabilities).

Disclaimer

test conducted this threat analysis using the open-source Threagile toolkit on the applications and systems that were modeled as of this report's date. Information security threats are continually changing, with new vulnerabilities discovered on a daily basis, and no application can ever be 100% secure no matter how much threat modeling is conducted. It is recommended to execute threat modeling and also penetration testing on a regular basis (for example yearly) to ensure a high ongoing level of security and constantly check for new attack vectors.

This report cannot and does not protect against personal or business loss as the result of use of the applications or systems described. test and the Threagile toolkit offers no warranties, representations or legal certifications concerning the applications or systems it tests. All software includes defects: nothing in this document is intended to represent or warrant that threat modeling was complete and without error, nor does this document represent or warrant that the architecture analyzed is suitable to task, free of other defects than reported, fully compliant with any industry standards, or fully compatible with any operating system, hardware, or other application. Threat modeling tries to analyze the modeled architecture without having access to a real working system and thus cannot and does not test the implementation for defects and vulnerabilities. These kinds of checks would only be possible with a separate code review and penetration test against a working system and not via a threat model.

By using the resulting information you agree that test and the Threagile toolkit shall be held harmless in any event.

This report is confidential and intended for internal, confidential use by the client. The recipient is obligated to ensure the highly confidential contents are kept secret. The recipient assumes responsibility for further distribution of this document.

In this particular project, a timebox approach was used to define the analysis effort. This means that the author allotted a prearranged amount of time to identify and document threats. Because of this, there is no guarantee that all possible threats and risks are discovered. Furthermore, the analysis applies to a snapshot of the current state of the modeled architecture (based on the architecture information provided by the customer) at the examination time.

Report Distribution

Distribution of this report (in full or in part like diagrams or risk findings) requires that this disclaimer as well as the chapter about the Threagile toolkit and method used is kept intact as part of the distributed report or referenced from the distributed parts.