



Threagile

Agile Threat Modeling

Threat Model Report Dashboard

18 August 2022

Tom Schmidt

Table of Contents

Results Overview

Management Summary	5
Impact Analysis of 49 Initial Risks in 25 Categories	6
Risk Mitigation	9
Impact Analysis of 42 Remaining Risks in 23 Categories	10
Application Overview	13
Data-Flow Diagram	14
Security Requirements	15
Abuse Cases	16
Tag Listing	17
STRIDE Classification of Identified Risks	18
Assignment by Function	22
RAA Analysis	26
Data Mapping	27
Out-of-Scope Assets: 2 Assets	28
Potential Model Failures: 12 / 12 Risks	29
Questions: 0 / 3 Questions	30

Risks by Vulnerability Category

Identified Risks by Vulnerability Category	31
Some Individual Risk Example: 2 / 2 Risks	32
XML External Entity (XXE): 1 / 1 Risk	34
Cross-Site Scripting (XSS): 2 / 2 Risks	36
Missing Cloud Hardening: 4 / 4 Risks	38
Missing File Validation: 1 / 1 Risk	41
Server-Side Request Forgery (SSRF): 1 / 1 Risk	43
Unencrypted Communication: 1 / 1 Risk	45
Unguarded Access From Internet: 2 / 2 Risks	47
Accidental Secret Leak: 1 / 1 Risk	49
Code Backdooring: 1 / 1 Risk	51
Container Base Image Backdooring: 1 / 1 Risk	54
Cross-Site Request Forgery (CSRF): 2 / 2 Risks	56
LDAP-Injection: 1 / 1 Risk	58
Missing Build Infrastructure: 1 / 1 Risk	60
Missing Identity Store: 1 / 1 Risk	62
Missing Vault (Secret Storage): 1 / 1 Risk	64
Missing Web Application Firewall (WAF): 2 / 2 Risks	66
Unchecked Deployment: 2 / 2 Risks	68

Unencrypted Technical Assets: 4 / 4 Risks	70
Unnecessary Data Transfer: 6 / 6 Risks	72
DoS-risky Access Across Trust-Boundary: 2 / 2 Risks	74
Unnecessary Data Asset: 2 / 2 Risks	76
Wrong Communication Link Content: 1 / 1 Risk	78
Missing Hardening: 0 / 3 Risks	80
Missing Two-Factor Authentication (2FA): 0 / 4 Risks	82

Risks by Technical Asset

Identified Risks by Technical Asset	84
Git Repository: 9 / 12 Risks	85
Backend Admin Client: 5 / 5 Risks	89
Apache Webserver: 9 / 11 Risks	92
Identity Provider: 5 / 7 Risks	96
Load Balancer: 7 / 7 Risks	99
Customer Web Client: out-of-scope	102
External Development Client: out-of-scope	104

Data Breach Probabilities by Data Asset

Identified Data Breach Probabilities by Data Asset	107
Company Application Code: 20 / 25 Risks	108
Customer Accounts: 21 / 25 Risks	109
Server Application Code: 23 / 28 Risks	110
Customer Access Keys: 0 / 0 Risks	112
Customer Infrastructure data: 0 / 0 Risks	113
Customer Organization: 0 / 0 Risks	114
Customer State Files: 0 / 0 Risks	115
Database Customizing and Dumps: 0 / 0 Risks	116
Marketing Material: 0 / 0 Risks	117

Trust Boundaries

Application Network	118
Auth Handling Environment	118
AWS VPC	118
Dev Network	118
Web DMZ	119

Shared Runtime

Puh	120
-----	-----

About Threagile

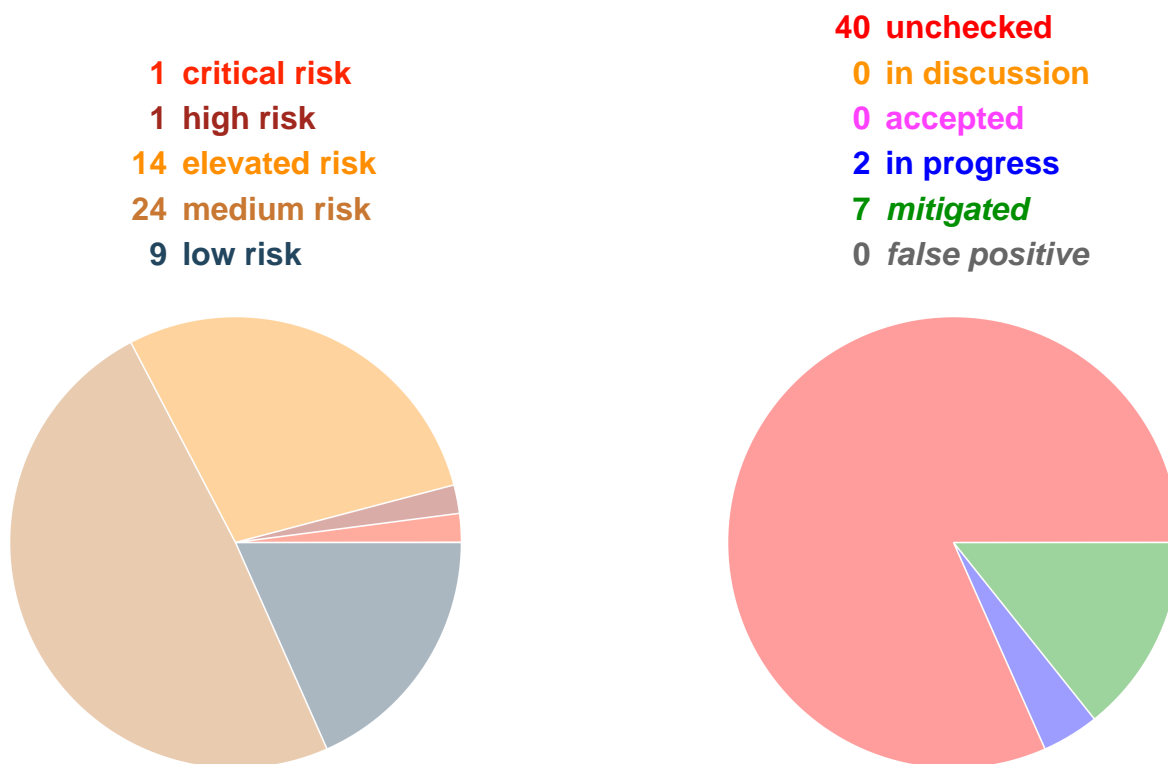
Risk Rules Checked by Threagile	121
Disclaimer	134

Management Summary

Threagile toolkit was used to model the architecture of "Dashboard" 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 "Dashboard" 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 **49 initial risks** in **25 categories** have been identified during the threat modeling process:



The Dashboard is the frontend for ... The Dashboard cumulates the functionality for Orchestrator, Marketplace and Policy Framework

Impact Analysis of 49 Initial Risks in 25 Categories

The most prevalent impacts of the **49 initial risks** (distributed over **25 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.

Critical: [Some Individual Risk Example](#): 2 Initial Risks - Exploitation likelihood is *Frequent* with *Very High* impact.

Some text describing the impact...

High: [XML External Entity \(XXE\)](#): 1 Initial Risk - Exploitation likelihood is *Very Likely* with *High* impact.

If this risk is unmitigated, attackers might be able to read sensitive files (configuration data, key/credential files, deployment files, business data files, etc.) from the filesystem of affected components and/or access sensitive services or files of other components.

Elevated: [Cross-Site Scripting \(XSS\)](#): 2 Initial Risks - Exploitation likelihood is *Likely* with *High* impact.

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

Elevated: [Missing Cloud Hardening](#): 4 Initial Risks - Exploitation likelihood is *Unlikely* with *Very High* impact.

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

Elevated: [Missing File Validation](#): 1 Initial Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to provide malicious files to the application.

Elevated: [Missing Hardening](#): 3 Initial Risks - Exploitation likelihood is *Likely* with *Medium* impact.

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

Elevated: [Server-Side Request Forgery \(SSRF\)](#): 1 Initial Risk - Exploitation likelihood is *Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to access sensitive services or files of network-reachable components by modifying outgoing calls of affected components.

Elevated: [Unencrypted Communication](#): 1 Initial Risk - Exploitation likelihood is *Likely* with *High* impact.

If this risk is unmitigated, network attackers might be able to eavesdrop on unencrypted sensitive data sent between components.

Elevated: [Unguarded Access From Internet](#): 2 Initial Risks - Exploitation likelihood is *Very Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to directly attack sensitive systems without any hardening components in-between due to them being directly exposed on the internet.

Medium: Accidental Secret Leak: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *High* impact. If this risk is unmitigated, attackers which have access to affected sourcecode repositories or artifact registries might find secrets accidentally checked-in.

Medium: Code Backdooring: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *High* impact. If this risk remains unmitigated, attackers might be able to execute code on and completely takeover production environments.

Medium: Container Base Image Backdooring: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *High* impact.

If this risk is unmitigated, attackers might be able to deeply persist in the target system by executing code in deployed containers.

Medium: Cross-Site Request Forgery (CSRF): 2 Initial Risks - Exploitation likelihood is *Very Likely* with *Low* impact.

If this risk remains unmitigated, attackers might be able to trick logged-in victim users into unwanted actions within the web application by visiting an attacker controlled web site.

Medium: LDAP-Injection: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *High* impact.

If this risk remains unmitigated, attackers might be able to modify LDAP queries and access more data from the LDAP server than allowed.

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 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 Two-Factor Authentication (2FA): 4 Initial Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to access or modify highly sensitive data without strong authentication.

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.

Medium: Missing Web Application Firewall (WAF): 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to apply standard attack pattern tests at great speed without any filtering.

Medium: **Unchecked Deployment:** 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk remains unmitigated, vulnerabilities in custom-developed software or their dependencies might not be identified during continuous deployment cycles.

Medium: **Unencrypted Technical Assets:** 4 Initial Risks - Exploitation likelihood is *Unlikely* with *High* impact.

If this risk is unmitigated, attackers might be able to access unencrypted data when successfully compromising sensitive components.

Medium: **Unnecessary Data Transfer:** 6 Initial Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to target unnecessarily transferred data.

Low: **DoS-risky Access Across Trust-Boundary:** 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

If this risk remains unmitigated, attackers might be able to disturb the availability of important parts of the system.

Low: **Unnecessary Data Asset:** 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

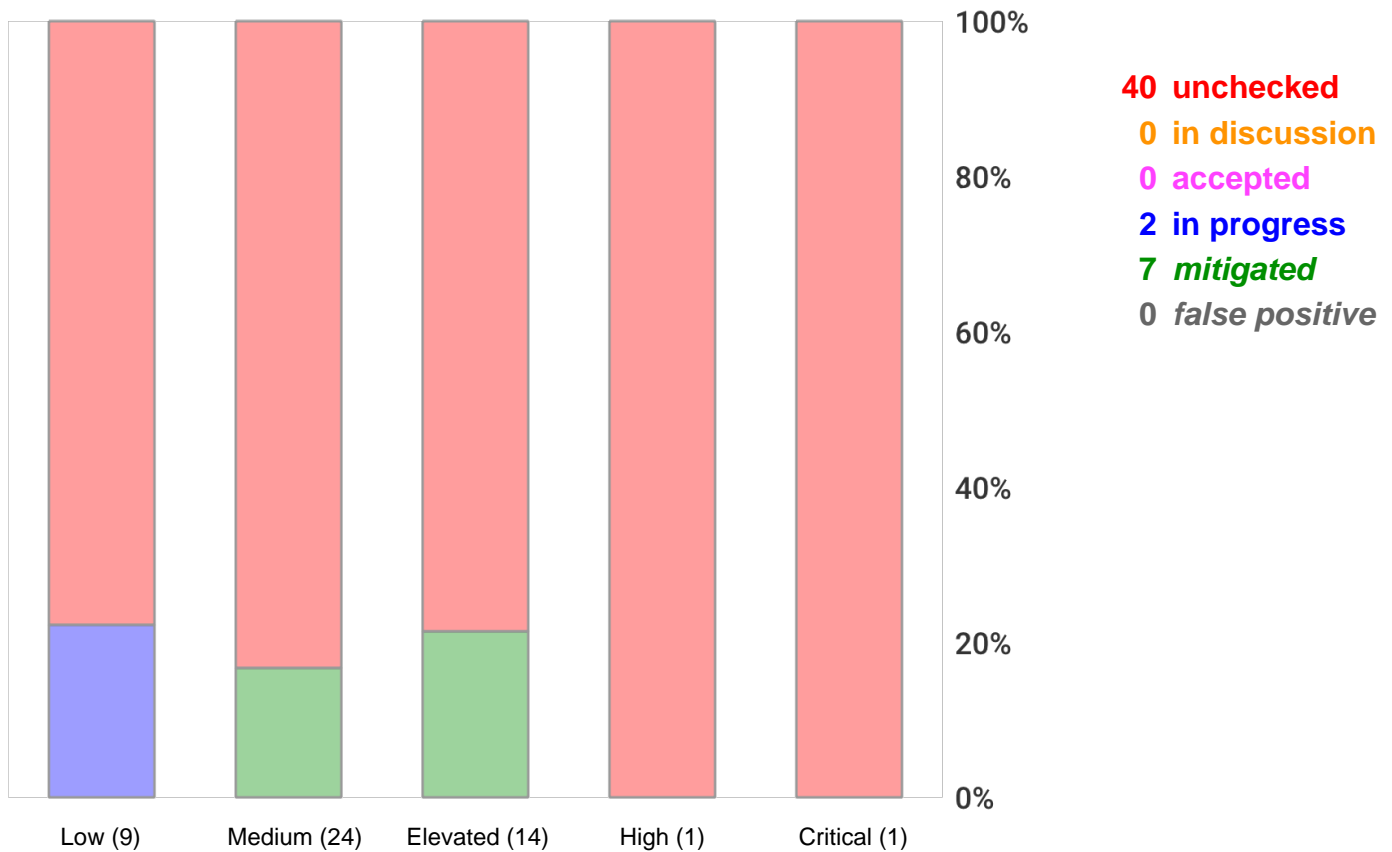
If this risk is unmitigated, attackers might be able to access unnecessary data assets using other vulnerabilities.

Low: **Wrong Communication Link Content:** 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Low* impact.

If this potential model error is not fixed, some risks might not be visible.

Risk Mitigation

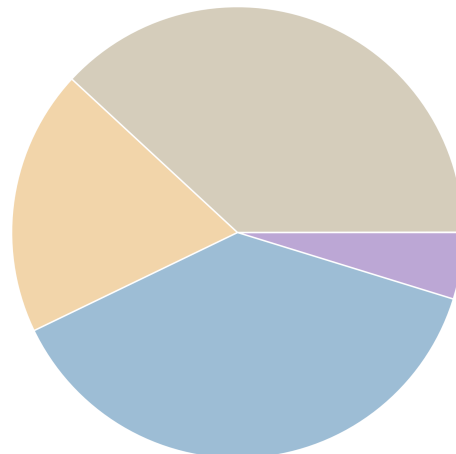
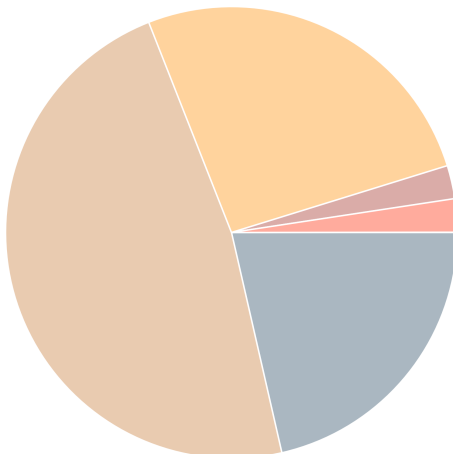
The following chart gives a high-level overview of the risk tracking status (including mitigated risks):



After removal of risks with status *mitigated* and *false positive* the following 42 remain unmitigated:

1 unmitigated critical risk
1 unmitigated high risk
11 unmitigated elevated risk
20 unmitigated medium risk
9 unmitigated low risk

2 business side related
16 architecture related
8 development related
16 operations related



Impact Analysis of 42 Remaining Risks in 23 Categories

The most prevalent impacts of the **42 remaining risks** (distributed over **23 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.

Critical: [Some Individual Risk Example](#): 2 Remaining Risks - Exploitation likelihood is *Frequent* with *Very High* impact.

Some text describing the impact...

High: [XML External Entity \(XXE\)](#): 1 Remaining Risk - Exploitation likelihood is *Very Likely* with *High* impact.

If this risk is unmitigated, attackers might be able to read sensitive files (configuration data, key/credential files, deployment files, business data files, etc.) from the filesystem of affected components and/or access sensitive services or files of other components.

Elevated: [Cross-Site Scripting \(XSS\)](#): 2 Remaining Risks - Exploitation likelihood is *Likely* with *High* impact.

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

Elevated: [Missing Cloud Hardening](#): 4 Remaining Risks - Exploitation likelihood is *Unlikely* with *Very High* impact.

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

Elevated: [Missing File Validation](#): 1 Remaining Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to provide malicious files to the application.

Elevated: [Server-Side Request Forgery \(SSRF\)](#): 1 Remaining Risk - Exploitation likelihood is *Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to access sensitive services or files of network-reachable components by modifying outgoing calls of affected components.

Elevated: [Unencrypted Communication](#): 1 Remaining Risk - Exploitation likelihood is *Likely* with *High* impact.

If this risk is unmitigated, network attackers might be able to eavesdrop on unencrypted sensitive data sent between components.

Elevated: [Unguarded Access From Internet](#): 2 Remaining Risks - Exploitation likelihood is *Very Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to directly attack sensitive systems without any hardening components in-between due to them being directly exposed on the internet.

Medium: [Accidental Secret Leak](#): 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *High* impact.

If this risk is unmitigated, attackers which have access to affected sourcecode repositories or artifact registries might find secrets accidentally checked-in.

Medium: **Code Backdooring**: 1 Remaining Risk - Exploitation likelihood is *Unlikely with High impact*.

If this risk remains unmitigated, attackers might be able to execute code on and completely takeover production environments.

Medium: **Container Base Image Backdooring**: 1 Remaining Risk - Exploitation likelihood is *Unlikely with High impact*.

If this risk is unmitigated, attackers might be able to deeply persist in the target system by executing code in deployed containers.

Medium: **Cross-Site Request Forgery (CSRF)**: 2 Remaining Risks - Exploitation likelihood is *Very Likely with Low impact*.

If this risk remains unmitigated, attackers might be able to trick logged-in victim users into unwanted actions within the web application by visiting an attacker controlled web site.

Medium: **LDAP-Injection**: 1 Remaining Risk - Exploitation likelihood is *Unlikely with High impact*.

If this risk remains unmitigated, attackers might be able to modify LDAP queries and access more data from the LDAP server than allowed.

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 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.

Medium: **Missing Web Application Firewall (WAF)**: 2 Remaining Risks - Exploitation likelihood is *Unlikely with Medium impact*.

If this risk is unmitigated, attackers might be able to apply standard attack pattern tests at great speed without any filtering.

Medium: **Unchecked Deployment**: 2 Remaining Risks - Exploitation likelihood is *Unlikely with Medium impact*.

If this risk remains unmitigated, vulnerabilities in custom-developed software or their dependencies might not be identified during continuous deployment cycles.

Medium: **Unencrypted Technical Assets**: 4 Remaining Risks - Exploitation likelihood is *Unlikely with High impact*.

If this risk is unmitigated, attackers might be able to access unencrypted data when successfully compromising sensitive components.

Medium: Unnecessary Data Transfer: 6 Remaining Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to target unnecessarily transferred data.

Low: DoS-risky Access Across Trust-Boundary: 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

If this risk remains unmitigated, attackers might be able to disturb the availability of important parts of the system.

Low: Unnecessary Data Asset: 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

If this risk is unmitigated, attackers might be able to access unnecessary data assets using other vulnerabilities.

Low: Wrong Communication Link Content: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Low* impact.

If this potential model error is not fixed, some risks might not be visible.

Application Overview

Business Criticality

The overall business criticality of "Dashboard" was rated as:

(archive | operational | important | critical | **MISSION-CRITICAL**)

Business Overview

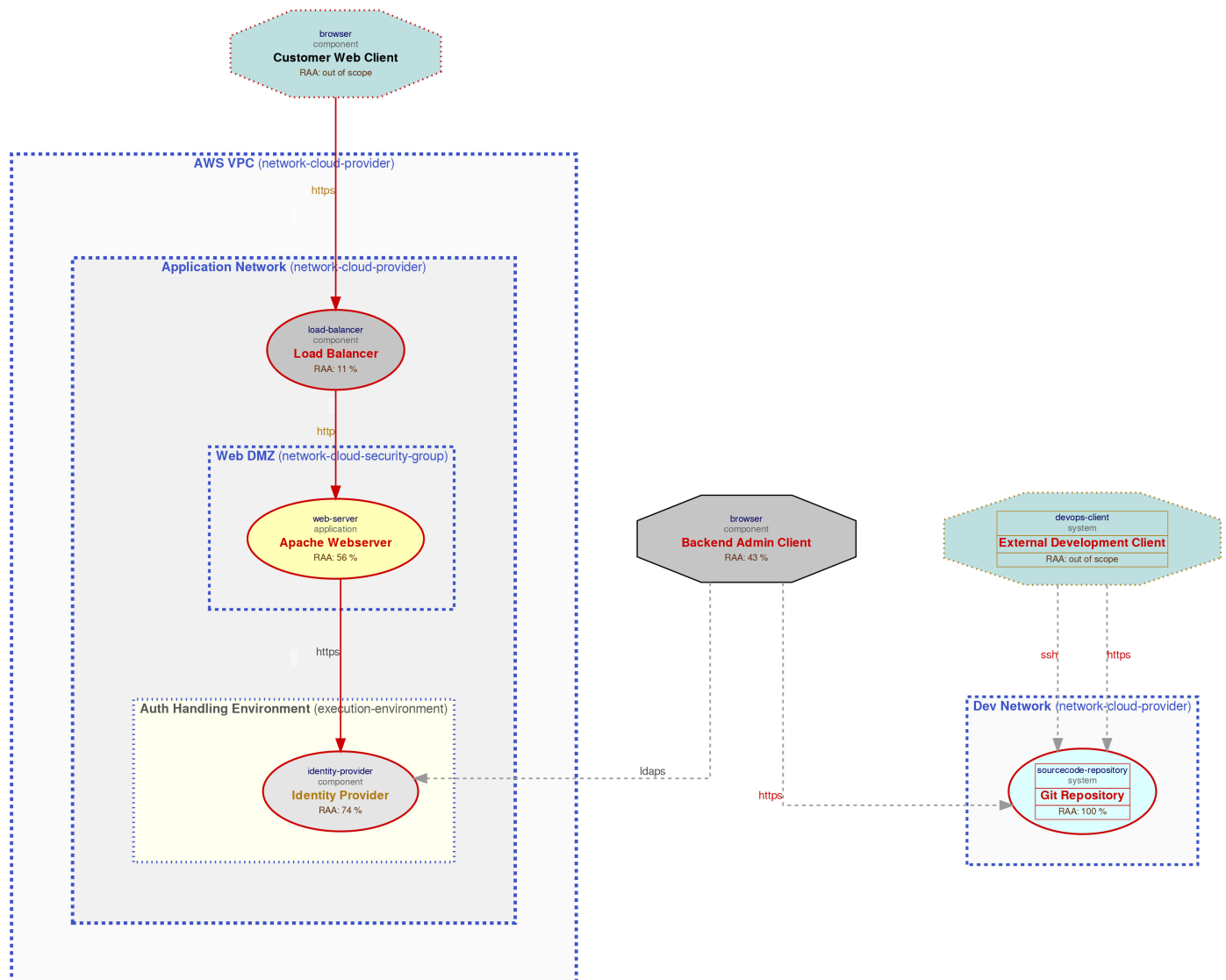
Currently in a very early stage

Technical Overview

The frontend is mainly build on react and some additional frameworks

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.

EU-DSGVO

Mandatory EU-Datenschutzgrundverordnung

Encryption

We need to apply strict encryption of customer sensitive data

Input Validation

Strict input validation is required to reduce the overall attack surface.

Securing Administrative Access

Administrative access must be secured with strong encryption and multi-factor authentication.

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.

Cross-Site Scripting Attacks

As a hacker I want to execute Cross-Site Scripting (XSS) and similar attacks in order to takeover victim sessions and cause reputational damage and eventually damage on the customer side.

Database Compromise

As a hacker I want to access the database backend of the Frontend in order to steal/modify sensitive business data.

Denial-of-Service

As a hacker I want to disturb the functionality of the backend system in order to cause indirect financial damage via unusable features. And attack the availability of the service as such.

Denial-of-Service of Enduser Functionality

As a hacker I want to disturb the functionality of the enduser parts of the application in order to cause direct financial damage (lower sales).

Identity Theft

As a hacker I want to steal identity data in order to reuse credentials and/or keys on other targets of the same company or outside.

PII Theft

As a hacker I want to steal PII (Personally Identifiable Information) data in order to blackmail the company and/or damage their repudiation by publishing them.

Ransomware

As a hacker I want to encrypt the storage and file systems in order to demand ransom.

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.

aws

Customer Traffic, Application Network, Dev Network

git

git-repo, Git Repository, Identity Provider

STRIDE Classification of Identified Risks

This chapter clusters and classifies the risks by STRIDE categories: In total **49 potential risks** have been identified during the threat modeling process of which **4 in the Spoofing** category, **17 in the Tampering** category, **2 in the Repudiation** category, **10 in the Information Disclosure** category, **2 in the Denial of Service** category, and **14 in the Elevation of Privilege** category.

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

Spoofing

Elevated: **Missing File Validation**: 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

When a technical asset accepts files, these input files should be strictly validated about filename and type.

Medium: **Cross-Site Request Forgery (CSRF)**: 2 / 2 Risks - Exploitation likelihood is *Very Likely* with *Low* impact.

When a web application is accessed via web protocols Cross-Site Request Forgery (CSRF) risks might arise.

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 *High* 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: **Missing Cloud Hardening**: 4 / 4 Risks - Exploitation likelihood is *Unlikely* with *Very High* impact.

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

Elevated: **Missing Hardening**: 0 / 3 Risks - Exploitation likelihood is *Likely* with *Medium* 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.

Medium: **Code Backdooring**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

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.

Medium: Container Base Image Backdooring: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

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.

Medium: LDAP-Injection: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

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.

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 Web Application Firewall (WAF): 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Medium: Unchecked Deployment: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Repudiation

Critical: Some Individual Risk Example: 2 / 2 Risks - Exploitation likelihood is *Frequent* with *Very High* impact.

Some text describing the risk category...

Information Disclosure

High: XML External Entity (XXE): 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *High* impact.

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

Elevated: Server-Side Request Forgery (SSRF): 1 / 1 Risk - Exploitation likelihood is *Likely* with *Medium* impact.

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.

Elevated: Unencrypted Communication: 1 / 1 Risk - Exploitation likelihood is *Likely* with *High* impact.

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

Medium: Accidental Secret Leak: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact. 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.

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.

Medium: Unencrypted Technical Assets: 4 / 4 Risks - Exploitation likelihood is *Unlikely* with *High* impact.

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.

Low: Wrong Communication Link Content: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Low* impact.

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.

Denial of Service

Low: DoS-risky Access Across Trust-Boundary: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

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

Elevation of Privilege

Elevated: Unguarded Access From Internet: 2 / 2 Risks - Exploitation likelihood is *Very Likely* with *Medium* impact.

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

Medium: Missing Two-Factor Authentication (2FA): 0 / 4 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Medium: Unnecessary Data Transfer: 6 / 6 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Low: Unnecessary Data Asset: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

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).

Assignment by Function

This chapter clusters and assigns the risks by functions which are most likely able to check and mitigate them: In total **49 potential risks** have been identified during the threat modeling process of which **6 should be checked by Business Side**, **16 should be checked by Architecture**, **8 should be checked by Development**, and **19 should be checked by Operations**.

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

Business Side

Critical: Some Individual Risk Example: 2 / 2 Risks - Exploitation likelihood is *Frequent* with *Very High* impact.

Some text describing the mitigation...

Medium: Missing Two-Factor Authentication (2FA): 0 / 4 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

Apply an authentication method to the technical asset protecting highly sensitive data via two-factor authentication for human users.

Architecture

Elevated: Unguarded Access From Internet: 2 / 2 Risks - Exploitation likelihood is *Very Likely* with *Medium* impact.

Encapsulate the asset behind a guarding service, application, or reverse-proxy. For admin maintenance a bastion-host should be used as a jump-server. For file transfer a store-and-forward-host should be used as an indirect file exchange platform.

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.).

Medium: Unchecked Deployment: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

Apply DevSecOps best-practices and use scanning tools to identify vulnerabilities in source- or byte-code, dependencies, container layers, and optionally also via dynamic scans against running test systems.

Medium: **Unnecessary Data Transfer**: 6 / 6 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

Try to avoid sending or receiving sensitive data assets which are not required (i.e. neither processed or stored) by the involved technical asset.

Low: **Unnecessary Data Asset**: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
Try to avoid having data assets that are not required/used.

Low: **Wrong Communication Link Content**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Low* impact.

Try to model the correct readonly flag and/or data sent/received of communication links. Also try to use communication link types matching the target technology/machine types.

Development

High: **XML External Entity (XXE)**: 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *High* impact.

Apply hardening of all XML parser instances in order to stay safe from XML External Entity (XXE) 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.

Elevated: **Cross-Site Scripting (XSS)**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *High* 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: **Missing File Validation**: 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

Filter by file extension and discard (if feasible) the name provided. Whitelist the accepted file types and determine the mime-type on the server-side (for example via "Apache Tika" or similar checks). If the file is retrievable by endusers and/or backoffice employees, consider performing scans for popular malware (if the files can be retrieved much later than they were uploaded, also apply a fresh malware scan during retrieval to scan with newer signatures of popular malware). Also enforce limits on maximum file size to avoid denial-of-service like scenarios.

Elevated: **Server-Side Request Forgery (SSRF)**: 1 / 1 Risk - Exploitation likelihood is *Likely* with *Medium* impact.

Try to avoid constructing the outgoing target URL with caller controllable values. Alternatively use a mapping (whitelist) when accessing outgoing URLs instead of creating them including caller controllable values. 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.

Medium: **Cross-Site Request Forgery (CSRF)**: 2 / 2 Risks - Exploitation likelihood is *Very Likely* with *Low* impact.

Try to use anti-CSRF tokens or the double-submit patterns (at least for logged-in requests). When your authentication scheme depends on cookies (like session or token cookies), consider

marking them with the same-site flag. 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.

Medium: LDAP-Injection: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

Try to use libraries that properly encode LDAP meta characters in searches and queries to access the LDAP sever in order to stay safe from LDAP-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

Elevated: Missing Cloud Hardening: 4 / 4 Risks - Exploitation likelihood is *Unlikely* with *Very 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).

Elevated: Missing Hardening: 0 / 3 Risks - Exploitation likelihood is *Likely* with *Medium* impact.

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

Elevated: Unencrypted Communication: 1 / 1 Risk - Exploitation likelihood is *Likely* with *High* impact.

Apply transport layer encryption to the communication link.

Medium: Accidental Secret Leak: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

Establish measures preventing accidental check-in or package-in of secrets into sourcecode repositories and artifact registries. This starts by using good .gitignore and .dockerignore files, but does not stop there. See for example tools like "*git-secrets*" or "*Talisman*" to have check-in preventive measures for secrets. Consider also to regularly scan your repositories for secrets accidentally checked-in using scanning tools like "*gitleaks*" or "*gitrob*".

Medium: Code Backdooring: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

Reduce the attack surface of backdooring the build pipeline by not directly exposing the build pipeline components on the public internet and also not exposing it in front of unmanaged (out-of-scope) developer clients. Also consider the use of code signing to prevent code modifications.

Medium: Container Base Image Backdooring: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *High* impact.

Apply hardening of all container infrastructures (see for example the *CIS-Benchmarks for Docker and Kubernetes* and the *Docker Bench for Security*). Use only trusted base images of the original vendors, verify digital signatures and apply image creation best practices. Also consider using Google's *Distroless* base images or otherwise very small base images. Regularly execute container image scans with tools checking the layers for vulnerable components.

Medium: **Missing Web Application Firewall (WAF): 2 / 2 Risks** - Exploitation likelihood is *Unlikely* with *Medium* impact.

Consider placing a Web Application Firewall (WAF) in front of the web-services and/or web-applications. For cloud environments many cloud providers offer pre-configured WAFs. Even reverse proxies can be enhanced by a WAF component via ModSecurity plugins.

Medium: **Unencrypted Technical Assets: 4 / 4 Risks** - Exploitation likelihood is *Unlikely* with *High* impact.

Apply encryption to the technical asset.

Low: **DoS-risky Access Across Trust-Boundary: 2 / 2 Risks** - Exploitation likelihood is *Unlikely* with *Low* impact.

Apply anti-DoS techniques like throttling and/or per-client load blocking with quotas. Also for maintenance access routes consider applying a VPN instead of public reachable interfaces. Generally applying redundancy on the targeted technical asset reduces the risk of DoS.

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.

Git Repository: RAA 100%

Git repository server

Identity Provider: RAA 74%

Identity provider server

Apache Webserver: RAA 56%

Apache Webserver hosting the API code and client-side code

Backend Admin Client: RAA 43%

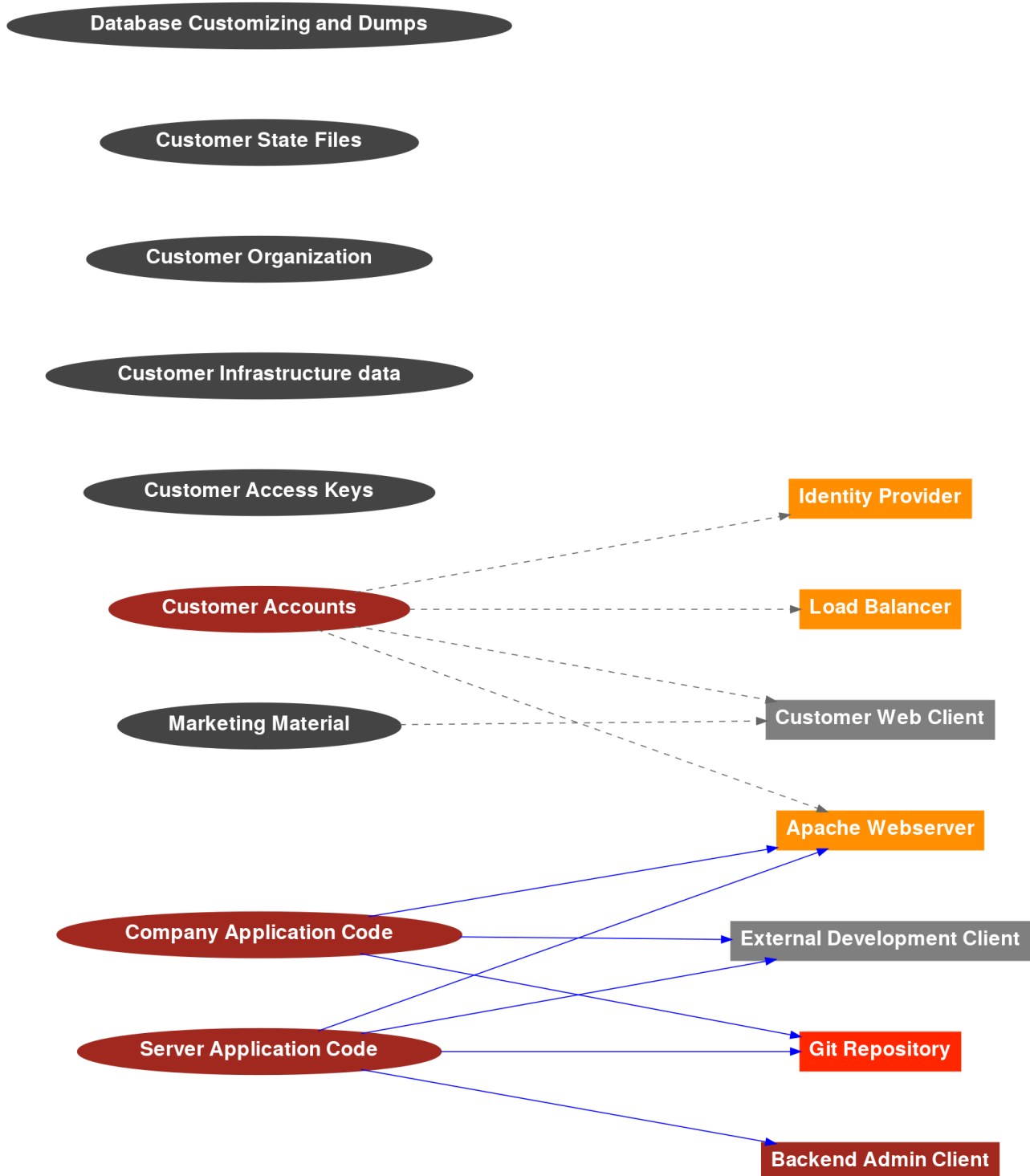
Backend admin client

Load Balancer: RAA 11%

Load Balancer (AWS-ALB)

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: 2 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.

Customer Web Client: out-of-scope

Owned and managed by enduser customer

External Development Client: out-of-scope

Owned and managed by external developers

Potential Model Failures: 12 / 12 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.

Medium: Unnecessary Data Transfer: 6 / 6 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Low: Unnecessary Data Asset: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

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).

Low: Wrong Communication Link Content: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Low* impact.

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.

Questions: 0 / 3 Questions

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

How are the admin clients managed/protected against compromise?

Crowdstrike, MFA

How are the build pipeline components managed/protected against compromise?

Good Question

How are the development clients managed/protected against compromise?

Crowdstrike, MFA

Identified Risks by Vulnerability Category

In total **49 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 1 as high, 14 as elevated, 24 as medium, and 9 as low.**

These risks are distributed across **25 vulnerability categories**. The following sub-chapters of this section describe each identified risk category.

Some Individual Risk Example: 2 / 2 Risks

Description (Repudiation): [CWE 693](#)

Some text describing the risk category...

Impact

Some text describing the impact...

Detection Logic

Some text describing the detection logic...

Risk Rating

Some text describing the risk assessment...

False Positives

Some text describing the most common types of false positives...

Mitigation (Business Side): Some text describing the action...

Some text describing the mitigation...

ASVS Chapter: [V0 - Something Strange](#)

Cheat Sheet: [example.com](#)

Check

Check if XYZ...

Risk Findings

The risk **Some Individual Risk Example** 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.

Critical Risk Severity

Example Individual Risk at Database: Exploitation likelihood is *Likely* with *Medium* impact.

something-strange@git-repo

Unchecked

Medium Risk Severity

Example Individual Risk at Contract Filesystem: Exploitation likelihood is *Frequent* with *Very High* impact.

something-strange@git-repo

Unchecked

XML External Entity (XXE): 1 / 1 Risk

Description (Information Disclosure): [CWE 611](#)

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

Impact

If this risk is unmitigated, attackers might be able to read sensitive files (configuration data, key/credential files, deployment files, business data files, etc.) from the filesystem of affected components and/or access sensitive services or files of other components.

Detection Logic

In-scope technical assets accepting XML data formats.

Risk 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).

False Positives

Fully trusted (i.e. cryptographically signed or similar) XML data can be considered as false positives after individual review.

Mitigation (Development): XML Parser Hardening

Apply hardening of all XML parser instances in order to stay safe from XML External Entity (XXE) 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: [V14 - Configuration Verification Requirements](#)

Cheat Sheet: [XML External Entity Prevention Cheat Sheet](#)

Check

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

Risk Findings

The risk **XML External Entity (XXE)** 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.

High Risk Severity

XML External Entity (XXE) risk at **Backend Admin Client**: Exploitation likelihood is *Very Likely* with *High* impact.

[xml-external-entity@backend-admin-client](#)

Unchecked

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 **Apache Webserver**: Exploitation likelihood is *Likely* with *High* impact.

[cross-site-scripting@apache-webserver](#)

Unchecked

Cross-Site Scripting (XSS) risk at **Identity Provider**: Exploitation likelihood is *Likely* with *High* impact.

[cross-site-scripting@identity-provider](#)

Unchecked

Missing Cloud Hardening: 4 / 4 Risks

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 **4 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

Missing Cloud Hardening (AWS) risk at Application Network: [CIS Benchmark for AWS](#): Exploitation likelihood is *Unlikely* with *Very High* impact.

[missing-cloud-hardening@application-network](#)

Unchecked

Missing Cloud Hardening (AWS) risk at Dev Network: [CIS Benchmark for AWS](#): Exploitation likelihood is *Unlikely* with *Very High* impact.

[missing-cloud-hardening@dev-network](#)

Unchecked

Missing Cloud Hardening risk at AWS VPC: Exploitation likelihood is *Unlikely* with *Very High* impact.

[missing-cloud-hardening@aws-vpc](#)

Unchecked

Missing Cloud Hardening risk at Web DMZ: Exploitation likelihood is *Unlikely* with *Very High* impact.

[missing-cloud-hardening@web-dmz](#)

Unchecked

Missing File Validation: 1 / 1 Risk

Description (Spoofing): [CWE 434](#)

When a technical asset accepts files, these input files should be strictly validated about filename and type.

Impact

If this risk is unmitigated, attackers might be able to provide malicious files to the application.

Detection Logic

In-scope technical assets with custom-developed code accepting file data formats.

Risk Rating

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

False Positives

Fully trusted (i.e. cryptographically signed or similar) files can be considered as false positives after individual review.

Mitigation (Development): File Validation

Filter by file extension and discard (if feasible) the name provided. Whitelist the accepted file types and determine the mime-type on the server-side (for example via "Apache Tika" or similar checks). If the file is retrievable by endusers and/or backoffice employees, consider performing scans for popular malware (if the files can be retrieved much later than they were uploaded, also apply a fresh malware scan during retrieval to scan with newer signatures of popular malware). Also enforce limits on maximum file size to avoid denial-of-service like scenarios.

ASVS Chapter: [V12 - File and Resources Verification Requirements](#)

Cheat Sheet: [File Upload Cheat Sheet](#)

Check

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

Risk Findings

The risk **Missing File Validation** 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

Missing File Validation risk at **Apache Webserver**: Exploitation likelihood is *Very Likely* with *Medium* impact.

[missing-file-validation@apache-webserver](#)

Unchecked

Server-Side Request Forgery (SSRF): 1 / 1 Risk

Description (Information Disclosure): [CWE 918](#)

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.

Impact

If this risk is unmitigated, attackers might be able to access sensitive services or files of network-reachable components by modifying outgoing calls of affected components.

Detection Logic

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

Risk 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.

False Positives

Servers not sending outgoing web requests can be considered as false positives after review.

Mitigation (Development): SSRF Prevention

Try to avoid constructing the outgoing target URL with caller controllable values. Alternatively use a mapping (whitelist) when accessing outgoing URLs instead of creating them including caller controllable values. 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: [V12 - File and Resources Verification Requirements](#)

Cheat Sheet: [Server Side Request Forgery Prevention Cheat Sheet](#)

Check

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

Risk Findings

The risk **Server-Side Request Forgery (SSRF)** 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

Server-Side Request Forgery (SSRF) risk at **Apache Webserver** server-side web-requesting the target **Identity Provider** via **Auth Credential Check Traffic**: Exploitation likelihood is *Likely* with *Medium* impact.

`server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic`

Unchecked

Unencrypted Communication: 1 / 1 Risk

Description (Information Disclosure): [CWE 319](#)

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

Impact

If this risk is unmitigated, network attackers might be able to to eavesdrop on unencrypted sensitive data sent between components.

Detection Logic

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.

Risk Rating

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

False Positives

When all sensitive data sent over the communication link is already fully encrypted on document or data level. Also intra-container/pod communication can be considered false positive when container orchestration platform handles encryption.

Mitigation (Operations): Encryption of Communication Links

Apply transport layer encryption to the communication link.

ASVS Chapter: [V9 - Communication Verification Requirements](#)

Cheat Sheet: [Transport Layer Protection Cheat Sheet](#)

Check

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

Risk Findings

The risk **Unencrypted Communication** 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

Unencrypted Communication named **Web Application Traffic** between **Load Balancer** and **Apache Webserver** transferring authentication data (like credentials, token, session-id, etc.):
Exploitation likelihood is *Likely* with *High* impact.

`unencrypted-communication@load-balancer>web-application-traffic@load-balancer@apache-webserver`

Unchecked

Unguarded Access From Internet: 2 / 2 Risks

Description (Elevation of Privilege): [CWE 501](#)

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

Impact

If this risk is unmitigated, attackers might be able to directly attack sensitive systems without any hardening components in-between due to them being directly exposed on the internet.

Detection Logic

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.

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.

False Positives

When other means of filtering client requests are applied equivalent of reverse-proxy, waf, or gateway components.

Mitigation (Architecture): Encapsulation of Technical Asset

Encapsulate the asset behind a guarding service, application, or reverse-proxy. For admin maintenance a bastion-host should be used as a jump-server. For file transfer a store-and-forward-host should be used as an indirect file exchange platform.

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 **Unguarded Access From Internet** 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

Unguarded Access from Internet of Git Repository by External Development Client via Git-Repo Code Write Access: Exploitation likelihood is *Very Likely* with *Medium* impact.

[unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-code-write-access](#)

Unchecked

Unguarded Access from Internet of Git Repository by External Development Client via Git-Repo Web-UI Access: Exploitation likelihood is *Very Likely* with *Medium* impact.

[unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-web-ui-access](#)

Unchecked

Accidental Secret Leak: 1 / 1 Risk

Description (Information Disclosure): [CWE 200](#)

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.

Impact

If this risk is unmitigated, attackers which have access to affected sourcecode repositories or artifact registries might find secrets accidentally checked-in.

Detection Logic

In-scope sourcecode repositories and artifact registries.

Risk Rating

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

False Positives

Usually no false positives.

Mitigation (Operations): Build Pipeline Hardening

Establish measures preventing accidental check-in or package-in of secrets into sourcecode repositories and artifact registries. This starts by using good `.gitignore` and `.dockerignore` files, but does not stop there. See for example tools like *"git-secrets"* or *"Talisman"* to have check-in preventive measures for secrets. Consider also to regularly scan your repositories for secrets accidentally checked-in using scanning tools like *"gitleaks"* or *"gitrob"*.

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 **Accidental Secret Leak** 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

Accidental Secret Leak (Git) risk at **Git Repository**: Git Leak Prevention: Exploitation likelihood is *Unlikely* with *High* impact.

accidental-secret-leak@git-repo

Unchecked

Code Backdooring: 1 / 1 Risk

Description (Tampering): [CWE 912](#)

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.

Impact

If this risk remains unmitigated, attackers might be able to execute code on and completely takeover production environments.

Detection Logic

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.

Risk 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.

False Positives

When the build-pipeline and sourcecode-repo is not exposed to the internet and considered fully trusted (which implies that all accessing clients are also considered fully trusted in terms of their patch management and applied hardening, which must be equivalent to a managed developer client environment) this can be considered a false positive after individual review.

Mitigation (Operations): Build Pipeline Hardening

Reduce the attack surface of backdooring the build pipeline by not directly exposing the build pipeline components on the public internet and also not exposing it in front of unmanaged (out-of-scope) developer clients. Also consider the use of code signing to prevent code modifications.

ASVS Chapter: [V10 - Malicious Code Verification Requirements](#)

Cheat Sheet: [Vulnerable Dependency Management Cheat Sheet](#)

Check

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

Risk Findings

The risk **Code Backdooring** 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

Code Backdooring risk at **Git Repository**: Exploitation likelihood is *Unlikely* with *High* impact.

[code-backdooring@git-repo](#)

Unchecked

Container Base Image Backdooring: 1 / 1 Risk

Description (Tampering): [CWE 912](#)

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.

See for example:

<https://techcrunch.com/2018/06/15/tainted-crypto-mining-containers-pulled-from-docker-hub/>

Impact

If this risk is unmitigated, attackers might be able to deeply persist in the target system by executing code in deployed containers.

Detection Logic

In-scope technical assets running as containers.

Risk Rating

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

False Positives

Fully trusted (i.e. reviewed and cryptographically signed or similar) base images of containers can be considered as false positives after individual review.

Mitigation (Operations): Container Infrastructure Hardening

Apply hardening of all container infrastructures (see for example the *CIS-Benchmarks for Docker and Kubernetes* and the *Docker Bench for Security*). Use only trusted base images of the original vendors, verify digital signatures and apply image creation best practices. Also consider using Google's *Distroless* base images or otherwise very small base images. Regularly execute container image scans with tools checking the layers for vulnerable components.

ASVS Chapter: [V10 - Malicious Code Verification Requirements](#)

Cheat Sheet: [Docker Security Cheat Sheet](#)

Check

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

Risk Findings

The risk **Container Base Image Backdooring** 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

Container Base Image Backdooring risk at **Apache Webserver**: Exploitation likelihood is *Unlikely* with *High* impact.

container-baseimage-backdooring@apache-webserver

Unchecked

Cross-Site Request Forgery (CSRF): 2 / 2 Risks

Description (Spoofing): [CWE 352](#)

When a web application is accessed via web protocols Cross-Site Request Forgery (CSRF) risks might arise.

Impact

If this risk remains unmitigated, attackers might be able to trick logged-in victim users into unwanted actions within the web application by visiting an attacker controlled web site.

Detection Logic

In-scope web applications accessed via typical web access protocols.

Risk Rating

The risk rating depends on the integrity rating of the data sent across the communication link.

False Positives

Web applications passing the authentication state via custom headers instead of cookies can eventually be false positives. Also when the web application 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): CSRF Prevention

Try to use anti-CSRF tokens or the double-submit patterns (at least for logged-in requests). When your authentication scheme depends on cookies (like session or token cookies), consider marking them with the same-site flag. 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: [V4 - Access Control Verification Requirements](#)

Cheat Sheet: [Cross-Site Request Forgery Prevention Cheat Sheet](#)

Check

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

Risk Findings

The risk **Cross-Site Request Forgery (CSRF)** 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

Cross-Site Request Forgery (CSRF) risk at Apache Webserver via Web Application Traffic from Load Balancer: Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@apache-webserver@load-balancer>web-application-traffic](#)

Unchecked

Cross-Site Request Forgery (CSRF) risk at Identity Provider via Auth Credential Check Traffic from Apache Webserver: Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@identity-provider@apache-webserver>auth-credential-check-traffic](#)

Unchecked

LDAP-Injection: 1 / 1 Risk

Description (Tampering): [CWE 90](#)

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.

Impact

If this risk remains unmitigated, attackers might be able to modify LDAP queries and access more data from the LDAP server than allowed.

Detection Logic

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

Risk Rating

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

False Positives

LDAP server queries by search values not consisting of parts controllable by the caller can be considered as false positives after individual review.

Mitigation (Development): LDAP-Injection Prevention

Try to use libraries that properly encode LDAP meta characters in searches and queries to access the LDAP sever in order to stay safe from LDAP-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: [LDAP Injection Prevention Cheat Sheet](#)

Check

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

Risk Findings

The risk **LDAP-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.

Medium Risk Severity

LDAP-Injection risk at Backend Admin Client against LDAP server Identity Provider via User Management Access: Exploitation likelihood is *Unlikely* with *High* impact.

`ldap-injection@backend-admin-client@identity-provider@backend-admin-client>user-management-access`

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 **Apache Webserver** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-build-infrastructure@apache-webserver](#)

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 **Load Balancer** as an example):
Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-identity-store@load-balancer](#)

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 **Git Repository** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-vault@git-repo](#)

Unchecked

Missing Web Application Firewall (WAF): 2 / 2 Risks

Description (Tampering): [CWE 1008](#)

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.

Impact

If this risk is unmitigated, attackers might be able to apply standard attack pattern tests at great speed without any filtering.

Detection Logic

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.

Risk Rating

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

False Positives

Targets only accessible via WAFs or reverse proxies containing a WAF component (like ModSecurity) can be considered as false positives after individual review.

Mitigation (Operations): Web Application Firewall (WAF)

Consider placing a Web Application Firewall (WAF) in front of the web-services and/or web-applications. For cloud environments many cloud providers offer pre-configured WAFs. Even reverse proxies can be enhanced by a WAF component via ModSecurity plugins.

ASVS Chapter: [V1 - Architecture, Design and Threat Modeling Requirements](#)

Cheat Sheet: [Virtual Patching Cheat Sheet](#)

Check

Is a Web Application Firewall (WAF) in place?

Risk Findings

The risk **Missing Web Application Firewall (WAF)** 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 Web Application Firewall (WAF) risk at **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-waf@apache-webserver](#)

Unchecked

Missing Web Application Firewall (WAF) risk at **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-waf@identity-provider](#)

Unchecked

Unchecked Deployment: 2 / 2 Risks

Description (Tampering): [CWE 1127](#)

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.

Impact

If this risk remains unmitigated, vulnerabilities in custom-developed software or their dependencies might not be identified during continuous deployment cycles.

Detection Logic

All development-relevant technical assets.

Risk Rating

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

False Positives

When the build-pipeline does not build any software components it can be considered a false positive after individual review.

Mitigation (Architecture): Build Pipeline Hardening

Apply DevSecOps best-practices and use scanning tools to identify vulnerabilities in source- or byte-code, dependencies, container layers, and optionally also via dynamic scans against running test systems.

ASVS Chapter: [V14 - Configuration Verification Requirements](#)

Cheat Sheet: [Vulnerable Dependency Management Cheat Sheet](#)

Check

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

Risk Findings

The risk **Unchecked Deployment** 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

Unchecked Deployment risk at **External Development Client**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unchecked-deployment@external-dev-client](#)

Unchecked

Low Risk Severity

Unchecked Deployment risk at **Git Repository**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unchecked-deployment@git-repo](#)

Unchecked

Unencrypted Technical Assets: 4 / 4 Risks

Description (Information Disclosure): [CWE 311](#)

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.

Impact

If this risk is unmitigated, attackers might be able to access unencrypted data when successfully compromising sensitive components.

Detection Logic

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.

Risk Rating

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

False Positives

When all sensitive data stored within the asset is already fully encrypted on document or data level.

Mitigation (Operations): Encryption of Technical Asset

Apply encryption to the technical asset.

ASVS Chapter: [V6 - Stored Cryptography Verification Requirements](#)

Cheat Sheet: [Cryptographic Storage Cheat Sheet](#)

Check

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

Risk Findings

The risk **Unencrypted Technical Assets** was found **4 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

Unencrypted Technical Asset named **Apache Webserver**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@apache-webserver](#)

Unchecked

Unencrypted Technical Asset named **Backend Admin Client**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@backend-admin-client](#)

Unchecked

Unencrypted Technical Asset named **Git Repository**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@git-repo](#)

Unchecked

Unencrypted Technical Asset named **Identity Provider**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@identity-provider](#)

Unchecked

Unnecessary Data Transfer: 6 / 6 Risks

Description (Elevation of Privilege): [CWE 1008](#)

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.

Impact

If this risk is unmitigated, attackers might be able to target unnecessarily transferred data.

Detection Logic

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.

Risk Rating

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

False Positives

Technical assets missing the model entries of either processing or storing the mentioned data assets can be considered as false positives (incomplete models) after individual review. These should then be addressed by completing the model so that all necessary data assets are processed and/or stored by the technical asset involved.

Mitigation (Architecture): Attack Surface Reduction

Try to avoid sending or receiving sensitive data assets which are not required (i.e. neither processed or stored) by the involved technical asset.

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 **Unnecessary Data Transfer** was found **6 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

Unnecessary Data Transfer of Customer Access Keys data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unnecessary-data-transfer@customer-access-keys@load-balancer@customer-client](#)

Unchecked

Unnecessary Data Transfer of Customer Accounts data at **Backend Admin Client** from/to **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unnecessary-data-transfer@customer-accounts@backend-admin-client@identity-provider](#)

Unchecked

Unnecessary Data Transfer of Customer Infrastructure data data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unnecessary-data-transfer@customer-infra@load-balancer@customer-client](#)

Unchecked

Low Risk Severity

Unnecessary Data Transfer of Company Application Code data at **Load Balancer** from/to **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unnecessary-data-transfer@client-application-code@load-balancer@apache-webserver](#)

Unchecked

Unnecessary Data Transfer of Company Application Code data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unnecessary-data-transfer@client-application-code@load-balancer@customer-client](#)

Unchecked

Unnecessary Data Transfer of Customer Organization data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unnecessary-data-transfer@customer-org@load-balancer@customer-client](#)

Unchecked

DoS-risky Access Across Trust-Boundary: 2 / 2 Risks

Description (Denial of Service): [CWE 400](#)

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

Impact

If this risk remains unmitigated, attackers might be able to disturb the availability of important parts of the system.

Detection Logic

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).

Risk 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.

False Positives

When the accessed target operations are not time- or resource-consuming.

Mitigation (Operations): Anti-DoS Measures

Apply anti-DoS techniques like throttling and/or per-client load blocking with quotas. Also for maintenance access routes consider applying a VPN instead of public reachable interfaces. Generally applying redundancy on the targeted technical asset reduces the risk of DoS.

ASVS Chapter: [V1 - Architecture, Design and Threat Modeling Requirements](#)

Cheat Sheet: [Denial of Service Cheat Sheet](#)

Check

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

Risk Findings

The risk **DoS-risky Access Across Trust-Boundary** 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.

Low Risk Severity

Denial-of-Service risky access of **Apache Webserver** by **Customer Web Client** via **Customer Traffic** forwarded via **Load Balancer**: Exploitation likelihood is *Unlikely* with *Low* impact.

`dos-risky-access-across-trust-boundary@apache-webserver@customer-client@customer-client>customer-traffic`

[in Progress](#) 2020-01-04 John Doe XYZ-1234
The hardening measures are being implemented and checked

Denial-of-Service risky access of **Identity Provider** by **Apache Webserver** via **Auth Credential Check Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

`dos-risky-access-across-trust-boundary@identity-provider@apache-webserver@apache-webserver>auth-credential-check-traffic`

[in Progress](#) 2020-01-04 John Doe XYZ-1234
The hardening measures are being implemented and checked

Unnecessary Data Asset: 2 / 2 Risks

Description (Elevation of Privilege): [CWE 1008](#)

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).

Impact

If this risk is unmitigated, attackers might be able to access unnecessary data assets using other vulnerabilities.

Detection Logic

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

Risk Rating

low

False Positives

Usually no false positives as this looks like an incomplete model.

Mitigation (Architecture): Attack Surface Reduction

Try to avoid having data assets that are not required/used.

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 **Unnecessary Data Asset** 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.

Low Risk Severity

Unnecessary Data Asset named **Customer State Files**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unnecessary-data-asset@customer-state](#)

Unchecked

Unnecessary Data Asset named **Database Customizing and Dumps**: Exploitation likelihood is *Unlikely* with *Low* impact.

[unnecessary-data-asset@db-dumps](#)

Unchecked

Wrong Communication Link Content: 1 / 1 Risk

Description (Information Disclosure): [CWE 1008](#)

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.

Impact

If this potential model error is not fixed, some risks might not be visible.

Detection Logic

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

Risk Rating

low

False Positives

Usually no false positives as this looks like an incomplete model.

Mitigation (Architecture): Model Consistency

Try to model the correct readonly flag and/or data sent/received of communication links. Also try to use communication link types matching the target technology/machine types.

ASVS Chapter: [V1 - Architecture, Design and Threat Modeling Requirements](#)

Cheat Sheet: [Threat Modeling Cheat Sheet](#)

Check

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

Risk Findings

The risk **Wrong Communication Link Content** 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.

Low Risk Severity

Wrong Communication Link Content (data assets sent/received not matching the communication link's readonly flag) at **Backend Admin Client** regarding communication link **git-repo**: Exploitation likelihood is *Unlikely* with *Low* impact.

wrong-communication-link-content@backend-admin-client@backend-admin-client>git-repo

Unchecked

Missing Hardening: 0 / 3 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 **3 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

Missing Hardening risk at **Apache Webserver**: Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@apache-webserver](#)

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Hardening risk at **Git Repository**: Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@git-repo](#)

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Hardening risk at **Identity Provider**: Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@identity-provider](#)

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Two-Factor Authentication (2FA): 0 / 4 Risks

Description (Elevation of Privilege): [CWE 308](#)

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.

Impact

If this risk is unmitigated, attackers might be able to access or modify highly sensitive data without strong authentication.

Detection Logic

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.

Risk Rating

medium

False Positives

Technical assets which do not process requests regarding functionality or data linked to end-users (customers) can be considered as false positives after individual review.

Mitigation (Business Side): Authentication with Second Factor (2FA)

Apply an authentication method to the technical asset protecting highly sensitive data via two-factor authentication for human users.

ASVS Chapter: [V2 - Authentication Verification Requirements](#)

Cheat Sheet: [Multifactor Authentication Cheat Sheet](#)

Check

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

Risk Findings

The risk **Missing Two-Factor Authentication (2FA)** was found **4 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 Two-Factor Authentication covering communication link **Git-Repo Code Write Access** from **External Development Client** to **Git Repository**: Exploitation likelihood is *Unlikely* with *Medium* impact.

`missing-authentication-second-factor@external-dev-client>git-repo-code-write-access@external-dev-client@git-repo`

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Two-Factor Authentication covering communication link **Git-Repo Web-UI Access** from **External Development Client** to **Git Repository**: Exploitation likelihood is *Unlikely* with *Medium* impact.

`missing-authentication-second-factor@external-dev-client>git-repo-web-ui-access@external-dev-client@git-repo`

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Two-Factor Authentication covering communication link **User Management Access** from **Backend Admin Client** to **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

`missing-authentication-second-factor@backend-admin-client>user-management-access@backend-admin-client@identity-provider`

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Missing Two-Factor Authentication covering communication link **Web Application Traffic** from **Customer Web Client** forwarded via **Load Balancer** to **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

`missing-authentication-second-factor@load-balancer>web-application-traffic@load-balancer@apache-webserver`

Mitigated 2020-01-04 John Doe XYZ-1234
The hardening measures were implemented and checked

Identified Risks by Technical Asset

In total **49 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 1 as high, 14 as elevated, 24 as medium, and 9 as low.**

These risks are distributed across **5 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.

Git Repository: 9 / 12 Risks

Description

Git repository server

Identified Risks of Asset

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

Critical Risk Severity

Example Individual Risk at Database: Exploitation likelihood is *Likely* with *Medium* impact.

something-strange@git-repo

Unchecked

Elevated Risk Severity

Unguarded Access from Internet of Git Repository by External Development Client via Git-Repo Code Write Access: Exploitation likelihood is *Very Likely* with *Medium* impact.

unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-code-write-access

Unchecked

Unguarded Access from Internet of Git Repository by External Development Client via Git-Repo Web-UI Access: Exploitation likelihood is *Very Likely* with *Medium* impact.

unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-web-ui-access

Unchecked

Missing Hardening risk at Git Repository: Exploitation likelihood is *Likely* with *Medium* impact.

missing-hardening@git-repo

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Medium Risk Severity

Example Individual Risk at Contract Filesystem: Exploitation likelihood is *Frequent* with *Very High* impact.

something-strange@git-repo

Unchecked

Accidental Secret Leak (Git) risk at Git Repository: Git Leak Prevention: Exploitation likelihood is *Unlikely* with *High* impact.

accidental-secret-leak@git-repo

Unchecked

Code Backdooring risk at **Git Repository**: Exploitation likelihood is *Unlikely* with *High* impact.

code-backdooring@git-repo

Unchecked

Unencrypted Technical Asset named **Git Repository**: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@git-repo

Unchecked

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

missing-vault@git-repo

Unchecked

Missing Two-Factor Authentication covering communication link **Git-Repo Code Write Access** from **External Development Client** to **Git Repository**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@external-dev-client>git-repo-code-write-access@external-dev-client@git-repo

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Missing Two-Factor Authentication covering communication link **Git-Repo Web-UI Access** from **External Development Client** to **Git Repository**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@external-dev-client>git-repo-web-ui-access@external-dev-client@git-repo

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Low Risk Severity

Unchecked Deployment risk at **Git Repository**: Exploitation likelihood is *Unlikely* with *Low* impact.

unchecked-deployment@git-repo

Unchecked

Asset Information

ID:	git-repo
Type:	process
Usage:	devops
RAA:	100 %
Size:	system
Technology:	sourcecode-repository
Tags:	git
Internet:	true

Machine:	virtual
Encryption:	none
Multi-Tenant:	true
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Company Application Code, Server Application Code
Data Stored:	Company Application Code, Server Application Code
Formats Accepted:	File

Asset Rating

Owner:	Company ABC
Confidentiality:	strictly-confidential (rated 5 in scale of 5)
Integrity:	critical (rated 4 in scale of 5)
Availability:	critical (rated 4 in scale of 5)
CIA-Justification:	The code repo pipeline might contain sensitive configuration values like backend credentials, certificates etc. and is therefore rated as strictly-confidential.

Incoming Communication Links: 3

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

Git-Repo Web-UI Access (incoming)

Link to the Git repo

Source:	External Development Client
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Company Application Code, Server Application Code
Data Sent:	Company Application Code, Server Application Code

Git-Repo Code Write Access (incoming)[Link to the Git repo](#)

Source:	External Development Client
Protocol:	ssh
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Company Application Code, Server Application Code
Data Sent:	Company Application Code, Server Application Code

git-repo (incoming)[Link to the Git repo \(web\)](#)

Source:	Backend Admin Client
Protocol:	https
Encrypted:	true
Authentication:	two-factor
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	git
VPN:	false
IP-Filtered:	false
Data Received:	none
Data Sent:	Server Application Code

Backend Admin Client: 5 / 5 Risks

Description

Backend admin client

Identified Risks of Asset

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

High Risk Severity

XML External Entity (XXE) risk at **Backend Admin Client**: Exploitation likelihood is *Very Likely* with *High* impact.

xml-external-entity@backend-admin-client

Unchecked

Medium Risk Severity

LDAP-Injection risk at **Backend Admin Client** against LDAP server **Identity Provider** via **User Management Access**: Exploitation likelihood is *Unlikely* with *High* impact.

ldap-injection@backend-admin-client@identity-provider@backend-admin-client>user-management-access

Unchecked

Unencrypted Technical Asset named **Backend Admin Client**: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@backend-admin-client

Unchecked

Unnecessary Data Transfer of Customer Accounts data at **Backend Admin Client** from/to **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unnecessary-data-transfer@customer-accounts@backend-admin-client@identity-provider

Unchecked

Low Risk Severity

Wrong Communication Link Content (data assets sent/received not matching the communication link's readonly flag) at **Backend Admin Client** regarding communication link **git-repo**: Exploitation likelihood is *Unlikely* with *Low* impact.

wrong-communication-link-content@backend-admin-client@backend-admin-client>git-repo

Unchecked

Asset Information

ID:	backend-admin-client
Type:	external-entity
Usage:	devops
RAA:	43 %
Size:	component
Technology:	browser
Tags:	none
Internet:	false
Machine:	physical
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	true
Data Processed:	Server Application Code
Data Stored:	Server Application Code
Formats Accepted:	File, JSON, XML

Asset Rating

Owner:	StackGuardian	
Confidentiality:	restricted	(rated 3 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	The client used by us to administer the system.	

Outgoing Communication Links: 2

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

git-repo (outgoing)

Link to the Git repo (web)

Target:	Git Repository
Protocol:	https
Encrypted:	true
Authentication:	two-factor
Authorization:	technical-user
Read-Only:	false
Usage:	devops

Tags: git
VPN: false
IP-Filtered: false
Data Sent: none
Data Received: Server Application Code

User Management Access (outgoing)

Link to the id for managing users

Target: Identity Provider
Protocol: Idaps
Encrypted: true
Authentication: credentials
Authorization: technical-user
Read-Only: false
Usage: devops
Tags: none
VPN: false
IP-Filtered: false
Data Sent: Customer Accounts
Data Received: Customer Accounts

Apache Webserver: 9 / 11 Risks

Description

Apache Webserver hosting the API code and client-side code

Identified Risks of Asset

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

Elevated Risk Severity

Cross-Site Scripting (XSS) risk at **Apache Webserver**: Exploitation likelihood is *Likely* with *High* impact.

cross-site-scripting@apache-webserver

Unchecked

Missing File Validation risk at **Apache Webserver**: Exploitation likelihood is *Very Likely* with *Medium* impact.

missing-file-validation@apache-webserver

Unchecked

Server-Side Request Forgery (SSRF) risk at **Apache Webserver** server-side web-requesting the target **Identity Provider** via **Auth Credential Check Traffic**: Exploitation likelihood is *Likely* with *Medium* impact.

server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic

Unchecked

Missing Hardening risk at **Apache Webserver**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-hardening@apache-webserver

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Medium Risk Severity

Container Base Image Backdooring risk at **Apache Webserver**: Exploitation likelihood is *Unlikely* with *High* impact.

container-baseimage-backdooring@apache-webserver

Unchecked

Unencrypted Technical Asset named **Apache Webserver**: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@apache-webserver

Unchecked

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

missing-build-infrastructure@apache-webserver

Unchecked

Missing Web Application Firewall (WAF) risk at **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-waf@apache-webserver

Unchecked

Cross-Site Request Forgery (CSRF) risk at **Apache Webserver** via **Web Application Traffic** from **Load Balancer**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@apache-webserver@load-balancer>web-application-traffic

Unchecked

Missing Two-Factor Authentication covering communication link **Web Application Traffic** from **Customer Web Client** forwarded via **Load Balancer** to **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@load-balancer>web-application-traffic@load-balancer@apache-webserver

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Low Risk Severity

Denial-of-Service risky access of **Apache Webserver** by **Customer Web Client** via **Customer Traffic** forwarded via **Load Balancer**: Exploitation likelihood is *Unlikely* with *Low* impact.

dos-risky-access-across-trust-boundary@apache-webserver@customer-client@customer-client>customer-traffic

in Progress

2020-01-04

John Doe

XYZ-1234

The hardening measures are being implemented and checked

Asset Information

ID:	apache-webserver
Type:	process
Usage:	business
RAA:	56 %
Size:	application
Technology:	web-server
Tags:	none
Internet:	false
Machine:	container
Encryption:	none
Multi-Tenant:	false
Redundant:	false

Custom-Developed: true
Client by Human: false
Data Processed: Company Application Code, Customer Accounts, Server Application Code
Data Stored: Company Application Code, Server Application Code
Formats Accepted: File, JSON

Asset Rating

Owner: Company ABC
Confidentiality: internal (rated 2 in scale of 5)
Integrity: critical (rated 4 in scale of 5)
Availability: critical (rated 4 in scale of 5)
CIA-Justification: The correct configuration and reachability of the web server is mandatory for all customer usages of the portal.

Outgoing Communication Links: 1

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

Auth Credential Check Traffic (outgoing)

Link to the identity provider server

Target: Identity Provider
Protocol: https
Encrypted: true
Authentication: credentials
Authorization: technical-user
Read-Only: false
Usage: business
Tags: none
VPN: false
IP-Filtered: false
Data Sent: Customer Accounts
Data Received: none

Incoming Communication Links: 1

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

Web Application Traffic (incoming)

Link to the web server

Source:	Load Balancer
Protocol:	http
Encrypted:	false
Authentication:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts
Data Sent:	Company Application Code, Customer Accounts

Identity Provider: 5 / 7 Risks

Description

Identity provider server

Identified Risks of Asset

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

Elevated Risk Severity

Cross-Site Scripting (XSS) risk at **Identity Provider**: Exploitation likelihood is *Likely* with *High* impact.

cross-site-scripting@identity-provider

Unchecked

Missing Hardening risk at **Identity Provider**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-hardening@identity-provider

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Medium Risk Severity

Unencrypted Technical Asset named **Identity Provider**: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@identity-provider

Unchecked

Missing Web Application Firewall (WAF) risk at **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-waf@identity-provider

Unchecked

Cross-Site Request Forgery (CSRF) risk at **Identity Provider** via **Auth Credential Check Traffic** from **Apache Webserver**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@identity-provider@apache-webserver>auth-credential-check-traffic

Unchecked

Missing Two-Factor Authentication covering communication link **User Management Access** from **Backend Admin Client** to **Identity Provider**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>user-management-access@backend-admin-client@identity-provider

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

Low Risk Severity

Denial-of-Service risky access of **Identity Provider** by **Apache Webserver** via **Auth Credential Check Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

dos-risky-access-across-trust-boundary@identity-provider@apache-webserver@apache-webserver>auth-credential-check-traffic

[in Progress](#)

2020-01-04

John Doe

XYZ-1234

The hardening measures are being implemented and checked

Asset Information

ID:	identity-provider
Type:	process
Usage:	business
RAA:	74 %
Size:	component
Technology:	identity-provider
Tags:	git
Internet:	false
Machine:	virtual
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Customer Accounts
Data Stored:	none
Formats Accepted:	File

Asset Rating

Owner:	Amazon Web Services	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	critical	(rated 4 in scale of 5)
Availability:	critical	(rated 4 in scale of 5)
CIA-Justification:	The auth data of the application	

Incoming Communication Links: 2

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

User Management Access (incoming)

[Link to the id for managing users](#)

Source:	Backend Admin Client
Protocol:	Idaps
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts
Data Sent:	Customer Accounts

Auth Credential Check Traffic (incoming)

[Link to the identity provider server](#)

Source:	Apache Webserver
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts
Data Sent:	none

Load Balancer: 7 / 7 Risks

Description

Load Balancer (AWS-ALB)

Identified Risks of Asset

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

Elevated Risk Severity

Unencrypted Communication named **Web Application Traffic** between **Load Balancer** and **Apache Webserver** transferring authentication data (like credentials, token, session-id, etc.): Exploitation likelihood is *Likely* with *High* impact.

unencrypted-communication@load-balancer>web-application-traffic@load-balancer@apache-webserver

Unchecked

Medium Risk Severity

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

missing-identity-store@load-balancer

Unchecked

Unnecessary Data Transfer of Customer Access Keys data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unnecessary-data-transfer@customer-access-keys@load-balancer@customer-client

Unchecked

Unnecessary Data Transfer of Customer Infrastructure data data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unnecessary-data-transfer@customer-infra@load-balancer@customer-client

Unchecked

Low Risk Severity

Unnecessary Data Transfer of Company Application Code data at **Load Balancer** from/to **Apache Webserver**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@client-application-code@load-balancer@apache-webserver

Unchecked

Unnecessary Data Transfer of Company Application Code data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@client-application-code@load-balancer@customer-client

Unchecked**Unnecessary Data Transfer of Customer Organization** data at **Load Balancer** from/to **Customer Web Client**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@customer-org@load-balancer@customer-client

Unchecked**Asset Information**

ID:	load-balancer
Type:	process
Usage:	business
RAA:	11 %
Size:	component
Technology:	load-balancer
Tags:	none
Internet:	false
Machine:	physical
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Customer Accounts
Data Stored:	none
Formats Accepted:	none of the special data formats accepted

Asset Rating

Owner:	Company ABC
Confidentiality:	internal (rated 2 in scale of 5)
Integrity:	mission-critical (rated 5 in scale of 5)
Availability:	mission-critical (rated 5 in scale of 5)
CIA-Justification:	The correct configuration and reachability of the load balancer is mandatory for all customer and Company XYZ usages of the portal and ERP system.

Outgoing Communication Links: 1

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

Web Application Traffic (outgoing)

Link to the web server

Target:	Apache Webserver
Protocol:	http
Encrypted:	false
Authentication:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Accounts
Data Received:	Company Application Code, Customer Accounts

Incoming Communication Links: 1

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

Customer Traffic (incoming)

Link to the load balancer

Source:	Customer Web Client
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	aws
VPN:	false
IP-Filtered:	false
Data Received:	Customer Access Keys, Customer Accounts
Data Sent:	Company Application Code, Customer Access Keys, Customer Accounts, Customer Infrastructure data, Customer Organization, Marketing Material

Customer Web Client: out-of-scope

Description

Customer Web Client

Identified Risks of Asset

Asset was defined as out-of-scope.

Asset Information

ID:	customer-client
Type:	external-entity
Usage:	business
RAA:	out-of-scope
Size:	component
Technology:	browser
Tags:	none
Internet:	true
Machine:	physical
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	true
Data Processed:	Customer Accounts, Marketing Material
Data Stored:	none
Formats Accepted:	none of the special data formats accepted

Asset Rating

Owner:	Customer	
Confidentiality:	internal	(rated 2 in scale of 5)
Integrity:	operational	(rated 2 in scale of 5)
Availability:	operational	(rated 2 in scale of 5)
CIA-Justification:	The client used by the customer to access the system.	

Asset Out-of-Scope Justification

Owned and managed by enduser customer

Outgoing Communication Links: 1

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

Customer Traffic (outgoing)

Link to the load balancer

Target:	Load Balancer
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	aws
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Access Keys, Customer Accounts
Data Received:	Company Application Code, Customer Access Keys, Customer Accounts, Customer Infrastructure data, Customer Organization, Marketing Material

External Development Client: out-of-scope

Description

External developer client

Identified Risks of Asset

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

Medium Risk Severity

Unchecked Deployment risk at **External Development Client**: Exploitation likelihood is *Unlikely with Medium impact*.

[unchecked-deployment@external-dev-client](#)

Unchecked

Asset Information

ID:	external-dev-client
Type:	external-entity
Usage:	devops
RAA:	out-of-scope
Size:	system
Technology:	devops-client
Tags:	none
Internet:	true
Machine:	physical
Encryption:	transparent
Multi-Tenant:	true
Redundant:	false
Custom-Developed:	false
Client by Human:	true
Data Processed:	Company Application Code, Server Application Code
Data Stored:	Company Application Code, Server Application Code
Formats Accepted:	File

Asset Rating

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

Integrity:	critical	(rated 4 in scale of 5)
Availability:	operational	(rated 2 in scale of 5)
CIA-Justification:	The clients used by external developers to create parts of the application code.	

Asset Out-of-Scope Justification

Owned and managed by external developers

Outgoing Communication Links: 2

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

Git-Repo Web-UI Access (outgoing)

[Link to the Git repo](#)

Target:	Git Repository
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Company Application Code, Server Application Code
Data Received:	Company Application Code, Server Application Code

Git-Repo Code Write Access (outgoing)

[Link to the Git repo](#)

Target:	Git Repository
Protocol:	ssh
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false

Usage: devops
Tags: none
VPN: false
IP-Filtered: false
Data Sent: Company Application Code, Server Application Code
Data Received: Company Application Code, Server Application Code

Identified Data Breach Probabilities by Data Asset

In total **49 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 1 as high, 14 as elevated, 24 as medium, and 9 as low.**

These risks are distributed across **9 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.

Company Application Code: 20 / 25 Risks

React and other client-side code delivered by the application.

ID:	client-application-code	
Usage:	devops	
Quantity:	very-few	
Tags:	none	
Origin:	StackGuardian	
Owner:	StackGuardian	
Confidentiality:	public	(rated 1 in scale of 5)
Integrity:	critical	(rated 4 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	The integrity of the public data is critical to avoid reputational damage and the availability is important on the long-term scale (but not critical) to keep the growth rate of the customer base steady.	
Processed by:	Apache Webserver, External Development Client, Git Repository	
Stored by:	Apache Webserver, External Development Client, Git Repository	
Sent via:	Git-Repo Web-UI Access, Git-Repo Code Write Access	
Received via:	Web Application Traffic, Git-Repo Web-UI Access, Git-Repo Code Write Access, Customer Traffic	

Data Breach: **probable**

Data Breach Risks: This data asset has data breach potential because of 20 remaining risks:

Probable: accidental-secret-leak@git-repo

Probable: code-backdooring@git-repo

Probable: container-baseimage-backdooring@apache-webserver

Probable: something-strange@git-repo

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@dev-network

Probable: missing-cloud-hardening@aws-vpc

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Possible: cross-site-scripting@apache-webserver

Possible: server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic

Possible: unchecked-deployment@external-dev-client

Possible: unchecked-deployment@git-repo

Possible: unencrypted-communication@load-balancer>web-application-traffic@load-balancer@apache-webserver

Possible: unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-code-write-access

Possible: unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-web-ui-access

Improbable: cross-site-request-forgery@apache-webserver@load-balancer>web-application-traffic

Improbable: missing-waf@apache-webserver

Improbable: unencrypted-asset@apache-webserver

Improbable: unencrypted-asset@git-repo

Customer Accounts: 21 / 25 Risks

Customer Accounts (including transient credentials when entered for checking them)

ID:	customer-accounts
Usage:	devops
Quantity:	many
Tags:	none
Origin:	Customer
Owner:	Customer Company
Confidentiality:	strictly-confidential (rated 5 in scale of 5)
Integrity:	important (rated 3 in scale of 5)
Availability:	critical (rated 4 in scale of 5)
CIA-Justification:	Customer account data for using the portal are required to be available to offer the portal functionality.
Processed by:	Apache Webserver, Customer Web Client, Identity Provider, Load Balancer
Stored by:	none
Sent via:	Web Application Traffic, User Management Access, Customer Traffic, Auth Credential Check Traffic
Received via:	Web Application Traffic, User Management Access, Customer Traffic
Data Breach:	probable
Data Breach Risks:	This data asset has data breach potential because of 21 remaining risks:

Probable: container-baseimage-backdooring@apache-webserver

Probable: ldap-injection@backend-admin-client@identity-provider@backend-admin-client>user-management-access

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@aws-vpc

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Possible: cross-site-scripting@apache-webserver

Possible: cross-site-scripting@identity-provider

Possible: server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic

Possible: unencrypted-communication@load-balancer>web-application-traffic@load-balancer@apache-webserver

Improbable: cross-site-request-forgery@apache-webserver@load-balancer>web-application-traffic

Improbable: cross-site-request-forgery@identity-provider@apache-webserver>auth-credential-check-traffic

Improbable: missing-waf@apache-webserver

Improbable: missing-waf@identity-provider

Improbable: unencrypted-asset@apache-webserver

Improbable: unencrypted-asset@identity-provider

Improbable: unnecessary-data-transfer@client-application-code@load-balancer@apache-webserver

Improbable: unnecessary-data-transfer@client-application-code@load-balancer@customer-client

Improbable: unnecessary-data-transfer@customer-access-keys@load-balancer@customer-client

Improbable: unnecessary-data-transfer@customer-infra@load-balancer@customer-client

Improbable: unnecessary-data-transfer@customer-org@load-balancer@customer-client

Server Application Code: 23 / 28 Risks

API and other server-side code of the application.

ID:	server-application-code	
Usage:	devops	
Quantity:	very-few	
Tags:	none	
Origin:	StackGuardian	
Owner:	StackGuardian	
Confidentiality:	restricted	(rated 3 in scale of 5)
Integrity:	mission-critical	(rated 5 in scale of 5)
Availability:	critical	(rated 4 in scale of 5)
CIA-Justification:	The integrity of the server side code is critical to avoid reputational damage and the availability is important on the long-term scale (but not critical) to keep the growth rate of the customer base steady.	
Processed by:	Apache Webserver, Backend Admin Client, External Development Client, Git Repository	
Stored by:	Apache Webserver, Backend Admin Client, External Development Client, Git Repository	
Sent via:	Git-Repo Web-UI Access, Git-Repo Code Write Access	
Received via:	git-repo, Git-Repo Web-UI Access, Git-Repo Code Write Access	
Data Breach:	probable	
Data Breach Risks:	This data asset has data breach potential because of 23 remaining risks:	

Probable: accidental-secret-leak@git-repo

Probable: code-backdooring@git-repo

Probable: container-baseimage-backdooring@apache-webserver

Probable: something-strange@git-repo

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@dev-network

Probable: missing-cloud-hardening@aws-vpc

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Probable: xml-external-entity@backend-admin-client

Possible: cross-site-scripting@apache-webserver

Possible: server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic

Possible: unchecked-deployment@external-dev-client

Possible: unchecked-deployment@git-repo

Possible: unencrypted-communication@load-balancer>web-application-traffic@load-balancer@apache-webserver

Possible: unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-code-write-access

Possible: unguarded-access-from-internet@git-repo@external-dev-client@external-dev-client>git-repo-web-ui-access

Improbable: cross-site-request-forgery@apache-webserver@load-balancer>web-application-traffic

Improbable: missing-waf@apache-webserver

Improbable: unencrypted-asset@apache-webserver

Improbable: unencrypted-asset@backend-admin-client

Improbable: unencrypted-asset@git-repo

Improbable: unnecessary-data-transfer@customer-accounts@backend-admin-client@identity-provider

Customer Access Keys: 0 / 0 Risks

Customer Access keys for various cloud accounts

ID:	customer-access-keys
Usage:	business
Quantity:	many
Tags:	none
Origin:	Customer
Owner:	Customer Company
Confidentiality:	strictly-confidential (rated 5 in scale of 5)
Integrity:	critical (rated 4 in scale of 5)
Availability:	mission-critical (rated 5 in scale of 5)
CIA-Justification:	No keys no cookies, if lost we are fucked
Processed by:	none
Stored by:	none
Sent via:	Customer Traffic
Received via:	Customer Traffic
Data Breach:	none
Data Breach Risks:	This data asset has no data breach potential.

Customer Infrastructure data: 0 / 0 Risks

Customer Infrastructure Data

ID:	customer-infra	
Usage:	business	
Quantity:	many	
Tags:	none	
Origin:	Customer	
Owner:	Customer Company	
Confidentiality:	strictly-confidential	(rated 5 in scale of 5)
Integrity:	critical	(rated 4 in scale of 5)
Availability:	critical	(rated 4 in scale of 5)
CIA-Justification:	Customer Infrastructure data will contain very sensitive and competitive data	
Processed by:	none	
Stored by:	none	
Sent via:	none	
Received via:	Customer Traffic	
Data Breach:	none	
Data Breach Risks:	This data asset has no data breach potential.	

Customer Organization: 0 / 0 Risks

Data asset being created as an Umbrella for individual accounts and assets in use

ID:	customer-org	
Usage:	devops	
Quantity:	many	
Tags:	none	
Origin:	Customer	
Owner:	Customer Company	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	mission-critical	(rated 5 in scale of 5)
CIA-Justification:	The Org Data asset consolidates users, workflows, configuration and keys	
Processed by:	none	
Stored by:	none	
Sent via:	none	
Received via:	Customer Traffic	
Data Breach:	none	
Data Breach Risks:	This data asset has no data breach potential.	

Customer State Files: 0 / 0 Risks

Customer State Files

ID:	customer-state
Usage:	business
Quantity:	many
Tags:	none
Origin:	Customer
Owner:	Customer Company
Confidentiality:	strictly-confidential (rated 5 in scale of 5)
Integrity:	mission-critical (rated 5 in scale of 5)
Availability:	mission-critical (rated 5 in scale of 5)
CIA-Justification:	The state files need to ensure integrity at all cost as this influences a core functionality of our platform
Processed by:	none
Stored by:	none
Sent via:	none
Received via:	none
Data Breach:	none
Data Breach Risks:	This data asset has no data breach potential.

Database Customizing and Dumps: 0 / 0 Risks

Data for customizing of the DB system, which might include full database dumps.

ID:	db-dumps
Usage:	devops
Quantity:	very-few
Tags:	none
Origin:	StackGuardian
Owner:	StackGuardian
Confidentiality:	strictly-confidential (rated 5 in scale of 5)
Integrity:	critical (rated 4 in scale of 5)
Availability:	critical (rated 4 in scale of 5)
CIA-Justification:	Data for customizing of the DB system, which might include full database dumps.
Processed by:	none
Stored by:	none
Sent via:	none
Received via:	none
Data Breach:	none
Data Breach Risks:	This data asset has no data breach potential.

Marketing Material: 0 / 0 Risks

Website and marketing data to inform potential customers and generate new leads.

ID:	marketing-material	
Usage:	business	
Quantity:	many	
Tags:	none	
Origin:		
Owner:		
Confidentiality:	public	(rated 1 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	operational	(rated 2 in scale of 5)
CIA-Justification:	public available data	
Processed by:	Customer Web Client	
Stored by:	none	
Sent via:	none	
Received via:	Customer Traffic	
Data Breach:	none	
Data Breach Risks:	This data asset has no data breach potential.	

Trust Boundaries

In total **5 trust boundaries** have been modeled during the threat modeling process.

AWS VPC

Amazon VPC

ID: aws-vpc
Type: [network-cloud-provider](#)
Tags: none
Assets inside: none
Boundaries nested: Application Network, Web DMZ

Application Network

Application Network

ID: application-network
Type: [network-cloud-provider](#)
Tags: aws
Assets inside: Load Balancer
Boundaries nested: Auth Handling Environment, Web DMZ

Auth Handling Environment

Auth Handling Environment

ID: auth-env
Type: execution-environment
Tags: none
Assets inside: Identity Provider
Boundaries nested: none

Dev Network

Development Network

ID: dev-network
Type: [network-cloud-provider](#)

Tags: aws
Assets inside: Git Repository
Boundaries nested: none

Web DMZ

Web DMZ

ID: web-dmz
Type: [network-cloud-security-group](#)
Tags: none
Assets inside: Apache Webserver
Boundaries nested: none

Shared Runtimes

In total **1 shared runtime** has been modeled during the threat modeling process.

Puh
blah

ID: puh
Tags: none
Assets running: none

Risk Rules Checked by Threagile

Threagile Version: 1.0.0

Threagile Build Timestamp: 20211121124511

Threagile Execution Timestamp: 20220818212356

Model Filename: /github/workspace/threagile.yaml

Model Hash (SHA256): 719688f6ff2c11fdb3132f1771572e7a3e4f6659394474e937975a68a91c1876

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:

Some Individual Risk Example

something-strange

Individual Risk Category

STRIDE: Repudiation
Description: Some text describing the risk category...
Detection: Some text describing the detection logic...
Rating: Some text describing the risk assessment...

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 webserver 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

Tom Schmidt 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. Tom Schmidt 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 Tom Schmidt 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.