



**Threagile**

Agile Threat Modeling

---

# Threat Model Report

## Example Application

11 November 2020

John Doe

# Table of Contents

## Results Overview

Management Summary	5
Impact Analysis of 84 Initial Risks in 28 Categories	6
Risk Mitigation	9
Impact Analysis of 59 Remaining Risks in 24 Categories	10
Application Overview	13
Data-Flow Diagram	14
Security Requirements	16
Abuse Cases	17
Tag Listing	19
STRIDE Classification of Identified Risks	21
Assignment by Function	25
RAA Analysis	29
Data Mapping	30
Out-of-Scope Assets: 4 Assets	31
Potential Model Failures: 3 / 3 Risks	32
Questions: 1 / 3 Questions	33

## Risks by Vulnerability Category

Identified Risks by Vulnerability Category	34
Some Individual Risk Example: 2 / 2 Risks	35
SQL/NoSQL-Injection: 1 / 1 Risk	37
XML External Entity (XXE): 1 / 1 Risk	39
Cross-Site Scripting (XSS): 4 / 4 Risks	41
Missing Authentication: 2 / 2 Risks	43
Missing Cloud Hardening: 5 / 5 Risks	45
Missing File Validation: 1 / 1 Risk	48
Path-Traversal: 1 / 1 Risk	50
Server-Side Request Forgery (SSRF): 2 / 2 Risks	52
Unencrypted Communication: 4 / 4 Risks	54
Unguarded Access From Internet: 3 / 3 Risks	56
Untrusted Deserialization: 2 / 2 Risks	58
Accidental Secret Leak: 1 / 1 Risk	60
Code Backdooring: 2 / 2 Risks	62
Container Base Image Backdooring: 2 / 2 Risks	65
Cross-Site Request Forgery (CSRF): 7 / 7 Risks	67
Missing Identity Propagation: 1 / 1 Risk	69
Missing Vault (Secret Storage): 1 / 1 Risk	71

Missing Web Application Firewall (WAF): 4 / 4 Risks	73
Mixed Targets on Shared Runtime: 1 / 1 Risk	75
Push instead of Pull Deployment: 2 / 2 Risks	77
Unchecked Deployment: 3 / 3 Risks	79
DoS-risky Access Across Trust-Boundary: 5 / 5 Risks	81
Missing Network Segmentation: 2 / 2 Risks	83
LDAP-Injection: 0 / 2 Risks	85
Missing Hardening: 0 / 6 Risks	87
Missing Two-Factor Authentication (2FA): 0 / 9 Risks	89
Unencrypted Technical Assets: 0 / 8 Risks	92

## Risks by Technical Asset

Identified Risks by Technical Asset	95
Customer Contract Database: 1 / 4 Risks	96
Backoffice ERP System: 15 / 19 Risks	99
Apache Webserver: 11 / 14 Risks	105
Contract Fileserver: 3 / 4 Risks	110
Git Repository: 5 / 8 Risks	113
Identity Provider: 4 / 7 Risks	117
Jenkins Buildserver: 5 / 8 Risks	120
Load Balancer: 1 / 1 Risk	124
Marketing CMS: 8 / 11 Risks	127
LDAP Auth Server: 1 / 3 Risks	132
Backend Admin Client: out-of-scope	135
Backoffice Client: out-of-scope	138
Customer Web Client: out-of-scope	140
External Development Client: out-of-scope	142

## Data Breach Probabilities by Data Asset

Identified Data Breach Probabilities by Data Asset	145
Build Job Config: 6 / 9 Risks	146
Client Application Code: 25 / 34 Risks	147
Customer Accounts: 39 / 57 Risks	149
Customer Contract Summaries: 7 / 8 Risks	151
Customer Contracts: 29 / 37 Risks	152
Customer Operational Data: 28 / 38 Risks	154
Database Customizing and Dumps: 6 / 9 Risks	156
ERP Customizing Data: 11 / 15 Risks	157
ERP Logs: 11 / 15 Risks	158
Marketing Material: 18 / 23 Risks	159

Server Application Code: 25 / 34 Risks	160
Some Internal Business Data: 28 / 38 Risks	162

**Trust Boundaries**

Application Network	164
Auth Handling Environment	164
Dev Network	164
ERP DMZ	164
Web DMZ	165

**Shared Runtime**

WebApp and Backoffice Virtualization	166
--------------------------------------	-----

**About Threagile**

Risk Rules Checked by Threagile	167
Disclaimer	180

## Management Summary

Threagile toolkit was used to model the architecture of "Example Application" 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 "Example Application" 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 **84 initial risks** in **28 categories** have been identified during the threat modeling process:



Just some **more** custom summary possible here...

# Impact Analysis of 84 Initial Risks in 28 Categories

The most prevalent impacts of the **84 initial risks** (distributed over **28 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: [SQL/NoSQL-Injection](#): 1 Initial Risk** - Exploitation likelihood is *Very Likely* with *High* impact.

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

**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 \(XSS\)](#): 4 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: [LDAP-Injection](#): 2 Initial Risks** - Exploitation likelihood is *Likely* 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.

**Elevated: [Missing Authentication](#): 2 Initial Risks** - Exploitation likelihood is *Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to access or modify sensitive data in an unauthenticated way.

**Elevated: [Missing Cloud Hardening](#): 5 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](#): 6 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: Path-Traversal:** 1 Initial Risk - Exploitation likelihood is *Very Likely* with *Medium* 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.

**Elevated: Server-Side Request Forgery (SSRF):** 2 Initial Risks - 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:** 4 Initial Risks - 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:** 3 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.

**Elevated: Untrusted Deserialization:** 2 Initial Risks - Exploitation likelihood is *Likely* with *Very High* impact.  
If this risk is unmitigated, attackers might be able to execute code on target systems by exploiting untrusted deserialization endpoints.

**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:** 2 Initial Risks - 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:** 2 Initial Risks - 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):** 7 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: Missing Identity Propagation:** 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.  
If this risk is unmitigated, attackers might be able to access or modify foreign data after a successful compromise of a component within the system due to missing resource-based authorization checks.

Medium: **Missing Two-Factor Authentication (2FA)**: 9 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)**: 4 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: **Mixed Targets on Shared Runtime**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are running on the same shared runtime.

Medium: **Push instead of Pull Deployment**: 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might have more potential target vectors for attacks, as the overall attack surface is unnecessarily increased.

Medium: **Unchecked Deployment**: 3 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**: 8 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.

Low: **DoS-risky Access Across Trust-Boundary**: 5 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: **Missing Network Segmentation**: 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are not separated by network segmentation.



## 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 **59** remain unmitigated:

- 1 unmitigated critical risk
- 2 unmitigated high risk
- 21 unmitigated elevated risk
- 27 unmitigated medium risk
- 8 unmitigated low risk

- 2 business side related
- 14 architecture related
- 17 development related
- 26 operations related



# Impact Analysis of 59 Remaining Risks in 24 Categories

The most prevalent impacts of the **59 remaining risks** (distributed over **24 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: [SQL/NoSQL-Injection](#): 1 Remaining Risk** - Exploitation likelihood is *Very Likely* with *High* impact.

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

**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 \(XSS\)](#): 4 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 Authentication](#): 2 Remaining Risks** - Exploitation likelihood is *Likely* with *Medium* impact.

If this risk is unmitigated, attackers might be able to access or modify sensitive data in an unauthenticated way.

**Elevated: [Missing Cloud Hardening](#): 5 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: [Path-Traversal](#): 1 Remaining Risk** - Exploitation likelihood is *Very Likely* with *Medium* 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.

**Elevated: Server-Side Request Forgery (SSRF): 2 Remaining Risks** - 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: 4 Remaining Risks** - 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: 3 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.

**Elevated: Untrusted Deserialization: 2 Remaining Risks** - Exploitation likelihood is *Likely with Very High impact*.

If this risk is unmitigated, attackers might be able to execute code on target systems by exploiting untrusted deserialization endpoints.

**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: 2 Remaining Risks** - 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: 2 Remaining Risks** - 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): 7 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: Missing Identity Propagation: 1 Remaining Risk** - Exploitation likelihood is *Unlikely with Medium impact*.

If this risk is unmitigated, attackers might be able to access or modify foreign data after a successful compromise of a component within the system due to missing resource-based authorization checks.

**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):** 4 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: Mixed Targets on Shared Runtime:** 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are running on the same shared runtime.

**Medium: Push instead of Pull Deployment:** 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

If this risk is unmitigated, attackers might have more potential target vectors for attacks, as the overall attack surface is unnecessarily increased.

**Medium: Unchecked Deployment:** 3 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.

**Low: DoS-risky Access Across Trust-Boundary:** 5 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: Missing Network Segmentation:** 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are not separated by network segmentation.

# Application Overview

## Business Criticality

The overall business criticality of "Example Application" was rated as:

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

## Business Overview

Some more *demo text* here and even images...

## Technical Overview

Some more *demo text* here and even images...

## 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.

## Data-Flow Diagram - Example Application



# Security Requirements

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

## **EU-DSGVO**

Mandatory EU-Datenschutzgrundverordnung

## **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.

## **CPU-Cycle Theft**

As a hacker I want to steal CPU cycles in order to transform them into money via installed crypto currency miners.

## **Contract Filesystem Compromise**

As a hacker I want to access the filesystem storing the contract PDFs in order to steal/modify contract data.

## **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.

## **Database Compromise**

As a hacker I want to access the database backend of the ERP-System 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.

## **Denial-of-Service of ERP/DB Functionality**

As a hacker I want to disturb the functionality of the ERP system and/or it's database in order to cause indirect financial damage via unusable internal ERP features (not related to customer portal).

## **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).

## **ERP-System Compromise**

As a hacker I want to access the ERP-System in order to steal/modify sensitive business data.

## **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.

## **apache**

Apache Webserver

## **aws**

Application Network

## **aws:ec2**

Apache Webserver

## **aws:s3**

Contract Fileserver

## **git**

Git Repository

## **jboss**

Identity Provider

## **jenkins**

Jenkins Buildserver

## **keycloak**

Identity Provider

## **linux**

Apache Webserver, Backoffice ERP System, Contract Fileserver, Customer Contract Database, External Development Client, Git Repository, Identity Provider, Jenkins Buildserver, LDAP Auth Server, Marketing CMS

## **mysql**

Customer Contract Database

**oracle**

Database Customizing and Dumps

**some-erp**

ERP Internal Access, ERP DMZ

**vmware**

WebApp and Backoffice Virtualization

# STRIDE Classification of Identified Risks

This chapter clusters and classifies the risks by STRIDE categories: In total **84 potential risks** have been identified during the threat modeling process of which **8 in the Spoofing** category, **33 in the Tampering** category, **2 in the Repudiation** category, **18 in the Information Disclosure** category, **5 in the Denial of Service** category, and **18 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)**: 7 / 7 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.

## Tampering

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

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

Elevated: **Cross-Site Scripting (XSS)**: 4 / 4 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: **LDAP-Injection**: 0 / 2 Risks - Exploitation likelihood is *Likely* 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.

Elevated: **Missing Cloud Hardening**: 5 / 5 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 / 6 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.

**Elevated: Untrusted Deserialization: 2 / 2 Risks** - Exploitation likelihood is *Likely with Very High impact*.

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

**Medium: Code Backdooring: 2 / 2 Risks** - 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: 2 / 2 Risks** - 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: Missing Web Application Firewall (WAF): 4 / 4 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: Push instead of Pull Deployment: 2 / 2 Risks** - Exploitation likelihood is *Unlikely with Medium impact*.

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.

**Medium: Unchecked Deployment: 3 / 3 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: Path-Traversal: 1 / 1 Risk** - Exploitation likelihood is *Very Likely* with *Medium* impact.

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.

**Elevated: Server-Side Request Forgery (SSRF): 2 / 2 Risks** - 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: 4 / 4 Risks** - 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: 0 / 8 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.

## Denial of Service

**Low: DoS-risky Access Across Trust-Boundary: 5 / 5 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: **Missing Authentication**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Medium* impact.

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

Elevated: **Unguarded Access From Internet**: 3 / 3 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 Identity Propagation**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.

Medium: **Missing Two-Factor Authentication (2FA)**: 0 / 9 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: **Mixed Targets on Shared Runtime**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

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

Low: **Missing Network Segmentation**: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.

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.



# Assignment by Function

This chapter clusters and assigns the risks by functions which are most likely able to check and mitigate them: In total **84 potential risks** have been identified during the threat modeling process of which **11 should be checked by Business Side**, **14 should be checked by Architecture**, **19 should be checked by Development**, and **40 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 / 9 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: Missing Authentication: 2 / 2 Risks** - Exploitation likelihood is *Likely* with *Medium* impact.

Apply an authentication method to the technical asset. To protect highly sensitive data consider the use of two-factor authentication for human users.

**Elevated: Unguarded Access From Internet: 3 / 3 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.

**Elevated: Untrusted Deserialization: 2 / 2 Risks** - Exploitation likelihood is *Likely* with *Very High* impact.

Try to avoid the deserialization of untrusted data (even of data within the same trust-boundary as long as it is sent across a remote connection) in order to stay safe from Untrusted Deserialization vulnerabilities. Alternatively a strict whitelisting approach of the classes/types/values to deserialize might help as well. 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: Missing Identity Propagation: 1 / 1 Risk** - Exploitation likelihood is *Unlikely* with *Medium* impact.

When processing requests for endusers if possible authorize in the backend against the propagated identity of the enduser. This can be achieved in passing JWTs or similar tokens and checking them in the backend services. For DevOps usages apply at least a technical-user authorization.

**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: Push instead of Pull Deployment:** 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Medium* impact.

Try to prefer pull-based deployments (like GitOps scenarios offer) over push-based deployments to reduce the attack surface of the production system.

**Medium: Unchecked Deployment:** 3 / 3 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.

## Development

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

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

**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):** 4 / 4 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: LDAP-Injection:** 0 / 2 Risks - Exploitation likelihood is *Likely* 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.

**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: Path-Traversal:** 1 / 1 Risk - Exploitation likelihood is *Very Likely* with *Medium* impact.

Before accessing the file cross-check that it resides in the expected folder and is of the expected type and filename/suffix. Try to use a mapping if possible instead of directly accessing by a filename which is (partly or fully) provided by the caller. 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: Server-Side Request Forgery (SSRF):** 2 / 2 Risks - 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):** 7 / 7 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.

## Operations

**Elevated: Missing Cloud Hardening:** 5 / 5 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 / 6 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:** 4 / 4 Risks - 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: 2 / 2 Risks** - 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: 2 / 2 Risks** - 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): 4 / 4 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: Mixed Targets on Shared Runtime: 1 / 1 Risk** - Exploitation likelihood is *Unlikely* with *Medium* impact.

Use separate runtime environments for running different target components or apply similar separation styles to prevent load- or breach-related problems originating from one more attacker-facing asset impacts also the other more critical rated backend/datastore assets.

**Medium: Unencrypted Technical Assets: 0 / 8 Risks** - Exploitation likelihood is *Unlikely* with *High* impact.

Apply encryption to the technical asset.

**Low: DoS-risky Access Across Trust-Boundary: 5 / 5 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.

**Low: Missing Network Segmentation: 2 / 2 Risks** - Exploitation likelihood is *Unlikely* with *Low* impact.

Apply a network segmentation trust-boundary around the highly sensitive assets and/or datastores.

# 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.

## **Backoffice ERP System: RAA 100%**

ERP system

## **Jenkins Buildserver: RAA 99%**

Jenkins buildserver

## **Apache Webserver: RAA 98%**

Apache Webserver hosting the API code and client-side code

## **Customer Contract Database: RAA 96%**

The database behind the ERP system

## **LDAP Auth Server: RAA 83%**

LDAP authentication server

## **Identity Provider: RAA 65%**

Identity provider server

## **Git Repository: RAA 48%**

Git repository server

## **Contract Fileserver: RAA 34%**

NFS Filesystem for storing the contract PDFs

## **Marketing CMS: RAA 34%**

CMS for the marketing content

## **Load Balancer: RAA 17%**

Load Balancer (HA-Proxy)

# 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: 4 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.

**Backend Admin Client:** out-of-scope

Owned and managed by ops provider

**Backoffice Client:** out-of-scope

Owned and managed by Company XYZ company

**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: 3 / 3 Risks

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

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

**Medium: Missing 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: Push instead of Pull Deployment: 2 / 2 Risks** - Exploitation likelihood is *Unlikely* with *Medium* impact.

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.



## Questions: 1 / 3 Questions

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

**How are the admin clients managed/protected against compromise?**

*- answer pending -*

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

*Managed by XYZ*

**How are the development clients managed/protected against compromise?**

*Managed by XYZ*

## Identified Risks by Vulnerability Category

In total **84 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 2 as high, 29 as elevated, 44 as medium, and 8 as low.**

These risks are distributed across **28 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@sql-database

**Unchecked**

### *Medium Risk Severity*

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

something-strange@contract-fileserver

**Unchecked**

## SQL/NoSQL-Injection: 1 / 1 Risk

**Description** (Tampering): [CWE 89](#)

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

### Impact

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

### Detection Logic

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

### Risk Rating

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

### False Positives

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

### Mitigation (Development): SQL/NoSQL-Injection Prevention

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

ASVS Chapter: [V5 - Validation, Sanitization and Encoding Verification Requirements](#)

Cheat Sheet: [SQL Injection Prevention Cheat Sheet](#)

### Check

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

## Risk Findings

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

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

### *High Risk Severity*

**SQL/NoSQL-Injection** risk at **Backoffice ERP System** against database **Customer Contract Database** via **Database Traffic**: Exploitation likelihood is *Very Likely* with *High* impact.

sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic

**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 **Backoffice ERP System**: Exploitation likelihood is *Very Likely* with *High* impact.

xml-external-entity@erp-system

**Unchecked**



## Cross-Site Scripting (XSS): 4 / 4 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 **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*

**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 Backoffice ERP System:** Exploitation likelihood is *Likely* with *High* impact.

[cross-site-scripting@erp-system](#)

**Unchecked**

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

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

**Unchecked**

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

[cross-site-scripting@marketing-cms](#)

**Unchecked**

## Missing Authentication: 2 / 2 Risks

**Description** (Elevation of Privilege): [CWE 306](#)

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

### Impact

If this risk is unmitigated, attackers might be able to access or modify sensitive data in an unauthenticated way.

### Detection Logic

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).

### Risk 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.

### 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 (Architecture): Authentication of Incoming Requests

Apply an authentication method to the technical asset. To protect highly sensitive data consider the use of two-factor authentication for human users.

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 Authentication** 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*

**Missing Authentication** covering communication link **CMS Content Traffic** from **Load Balancer** to **Marketing CMS**: Exploitation likelihood is *Likely* with *Medium* impact.

`missing-authentication@load-balancer>cms-content-traffic@load-balancer@marketing-cms`

**Unchecked**

**Missing Authentication** covering communication link **NFS Filesystem Access** from **Backoffice ERP System** to **Contract Fileserver**: Exploitation likelihood is *Likely* with *Medium* impact.

`missing-authentication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver`

**Unchecked**

## Missing Cloud Hardening: 5 / 5 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 **5 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 (EC2)** risk at **Apache Webserver**: [CIS Benchmark for Amazon Linux](#): Exploitation likelihood is *Unlikely* with *Very High* impact.

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

**Unchecked**

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

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

**Unchecked**

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

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

**Unchecked**

### *Medium Risk Severity*

**Missing Cloud Hardening (S3)** risk at **Contract Fileserver**: [Security Best Practices for AWS S3](#): Exploitation likelihood is *Unlikely* with *High* impact.

[missing-cloud-hardening@contract-fileserver](#)

**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**

## Path-Traversal: 1 / 1 Risk

**Description** (Information Disclosure): [CWE 22](#)

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.

### 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.

### Detection Logic

Filesystems accessed by in-scope callers.

### Risk Rating

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

### False Positives

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

### Mitigation (Development): Path-Traversal Prevention

Before accessing the file cross-check that it resides in the expected folder and is of the expected type and filename/suffix. Try to use a mapping if possible instead of directly accessing by a filename which is (partly or fully) provided by the caller. 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: [Input Validation Cheat Sheet](#)

### Check

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

## Risk Findings

The risk **Path-Traversal** 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*

**Path-Traversal** risk at **Backoffice ERP System** against filesystem **Contract Fileserver** via **NFS Filesystem Access**: Exploitation likelihood is *Very Likely* with *Medium* impact.

`path-traversal@erp-system@contract-fileserver@erp-system>nfs-filesystem-access`

**Unchecked**

## Server-Side Request Forgery (SSRF): 2 / 2 Risks

**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 **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*

**Server-Side Request Forgery (SSRF)** risk at **Apache Webserver** server-side web-requesting the target **Backoffice ERP System** via **ERP System Traffic**: Exploitation likelihood is *Likely* with *Medium* impact.

`server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic`

**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**

## Unencrypted Communication: 4 / 4 Risks

**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 **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

**Unencrypted Communication** named **Auth Traffic** between **Marketing CMS** and **LDAP Auth Server** transferring authentication data (like credentials, token, session-id, etc.): Exploitation likelihood is *Likely* with *High* impact.

[unencrypted-communication@marketing-cms>auth-traffic@marketing-cms@ldap-auth-server](#)

**Unchecked**

**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

**Unencrypted Communication** named **Database Traffic** between **Backoffice ERP System** and **Customer Contract Database** transferring authentication data (like credentials, token, session-id, etc.): Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-communication@erp-system>database-traffic@erp-system@sql-database](#)

**Unchecked**

**Unencrypted Communication** named **NFS Filesystem Access** between **Backoffice ERP System** and **Contract Fileserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unencrypted-communication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver](#)

**Unchecked**

## Unguarded Access From Internet: 3 / 3 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 **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*

**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**

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

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

**Unchecked**

## Untrusted Deserialization: 2 / 2 Risks

**Description** (Tampering): [CWE 502](#)

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

See <https://christian-schneider.net/JavaDeserializationSecurityFAQ.html> for more details.

### Impact

If this risk is unmitigated, attackers might be able to execute code on target systems by exploiting untrusted deserialization endpoints.

### Detection Logic

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

### 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) data deserialized can be considered as false positives after individual review.

### Mitigation (Architecture): Prevention of Deserialization of Untrusted Data

Try to avoid the deserialization of untrusted data (even of data within the same trust-boundary as long as it is sent across a remote connection) in order to stay safe from Untrusted Deserialization vulnerabilities. Alternatively a strict whitelisting approach of the classes/types/values to deserialize might help as well. 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: [Deserialization Cheat Sheet](#)

### Check

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

Risk Findings

The risk **Untrusted Deserialization** 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

**Untrusted Deserialization risk at Jenkins Buildserver:** Exploitation likelihood is *Likely* with *Very High* impact.

[untrusted-deserialization@jenkins-buildserver](#)

Unchecked

**Untrusted Deserialization risk at Backoffice ERP System:** Exploitation likelihood is *Likely* with *Very High* impact.

[untrusted-deserialization@erp-system](#)

Accepted	2020-01-04	John Doe	XYZ-1234
Risk accepted as tolerable			

## 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: 2 / 2 Risks

### 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 **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

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

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

**Unchecked**

**Code Backdooring** risk at **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *High* impact.

[code-backdooring@jenkins-buildserver](#)

**Unchecked**



## Container Base Image Backdooring: 2 / 2 Risks

**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 **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*

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

[container-baseimage-backdooring@apache-webserver](#)

**Unchecked**

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

[container-baseimage-backdooring@marketing-cms](#)

**Unchecked**

## Cross-Site Request Forgery (CSRF): 7 / 7 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 **7 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 Backoffice ERP System via ERP Internal Access from Backoffice Client:** Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@erp-system@backoffice-client>erp-internal-access](#)

**Unchecked**

**Cross-Site Request Forgery (CSRF) risk at Backoffice ERP System via ERP System Traffic from Apache Webserver:** Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@erp-system@apache-webserver>erp-system-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**

**Cross-Site Request Forgery (CSRF) risk at Marketing CMS via CMS Content Traffic from Load Balancer:** Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@marketing-cms@load-balancer>cms-content-traffic](#)

**Unchecked**

**Cross-Site Request Forgery (CSRF) risk at Marketing CMS via Marketing CMS Editing from Backoffice Client:** Exploitation likelihood is *Very Likely* with *Low* impact.

[cross-site-request-forgery@marketing-cms@backoffice-client>marketing-cms-editing](#)

**Unchecked**

**Cross-Site Request Forgery (CSRF) risk at Backoffice ERP System via ERP Web Access from Backend Admin Client:** Exploitation likelihood is *Likely* with *Low* impact.

[cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access](#)

**Unchecked**

## Missing Identity Propagation: 1 / 1 Risk

**Description** (Elevation of Privilege): [CWE 284](#)

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.

### Impact

If this risk is unmitigated, attackers might be able to access or modify foreign data after a successful compromise of a component within the system due to missing resource-based authorization checks.

### Detection Logic

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.

### Risk Rating

The risk rating (medium or high) depends on the confidentiality, integrity, and availability rating of the technical asset.

### 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** (Architecture): Identity Propagation and Resource-based Authorization

When processing requests for endusers if possible authorize in the backend against the propagated identity of the enduser. This can be achieved in passing JWTs or similar tokens and checking them in the backend services. For DevOps usages apply at least a technical-user authorization.

ASVS Chapter: [V4 - Access Control Verification Requirements](#)

Cheat Sheet: [Access Control Cheat Sheet](#)

### Check

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

## Risk Findings

The risk **Missing Identity Propagation** 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 Enduser Identity Propagation** over communication link **ERP System Traffic** from **Apache Webserver** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system](#)

**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 **Backoffice ERP System** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-vault@erp-system](#)

**Unchecked**



## Missing Web Application Firewall (WAF): 4 / 4 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 **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 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 **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[missing-waf@erp-system](#)

**Unchecked**

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

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

**Unchecked**

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

[missing-waf@marketing-cms](#)

**Unchecked**

## Mixed Targets on Shared Runtime: 1 / 1 Risk

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

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

### Impact

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are running on the same shared runtime.

### Detection Logic

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.

### 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.

### False Positives

When all assets running on the shared runtime are hardened and protected to the same extend as if all were containing/processing highly sensitive data.

### Mitigation (Operations): Runtime Separation

Use separate runtime environments for running different target components or apply similar separation styles to prevent load- or breach-related problems originating from one more attacker-facing asset impacts also the other more critical rated backend/datastore assets.

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 **Mixed Targets on Shared Runtime** 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*

**Mixed Targets on Shared Runtime** named **WebApp and Backoffice Virtualization** might enable attackers moving from one less valuable target to a more valuable one: Exploitation likelihood is *Unlikely* with *Medium* impact.

[mixed-targets-on-shared-runtime@webapp-virtualization](#)

**Unchecked**

## Push instead of Pull Deployment: 2 / 2 Risks

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

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.

### Impact

If this risk is unmitigated, attackers might have more potential target vectors for attacks, as the overall attack surface is unnecessarily increased.

### Detection Logic

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

### Risk Rating

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

### False Positives

Communication links that are not deployment paths can be considered as false positives after individual review.

**Mitigation** (Architecture): Build Pipeline Hardening

Try to prefer pull-based deployments (like GitOps scenarios offer) over push-based deployments to reduce the attack surface of the production system.

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 **Push instead of Pull 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*

**Push instead of Pull Deployment at Apache Webserver** via build pipeline asset **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[push-instead-of-pull-deployment@jenkins-buildserver](#)

**Unchecked**

**Push instead of Pull Deployment at Marketing CMS** via build pipeline asset **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[push-instead-of-pull-deployment@jenkins-buildserver](#)

**Unchecked**

## Unchecked Deployment: 3 / 3 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 **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.

### Medium Risk Severity

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

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

**Unchecked**

**Unchecked Deployment** risk at **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

[unchecked-deployment@jenkins-buildserver](#)

**Unchecked**

### Low Risk Severity

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

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

**Unchecked**



## DoS-risky Access Across Trust-Boundary: 5 / 5 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 **5 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 **Backoffice ERP System** by **Apache Webserver** via **ERP System Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

`dos-risky-access-across-trust-boundary@erp-system@apache-webserver@apache-webserver>erp-system-traffic`

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

**Denial-of-Service** risky access of **Backoffice ERP System** by **Backoffice Client** via **ERP Internal Access**: Exploitation likelihood is *Unlikely* with *Low* impact.

`dos-risky-access-across-trust-boundary@erp-system@backoffice-client@backoffice-client>erp-internal-access`

[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

**Denial-of-Service** risky access of **LDAP Auth Server** by **Marketing CMS** via **Auth Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

`dos-risky-access-across-trust-boundary@ldap-auth-server@marketing-cms@marketing-cms>auth-traffic`

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

## Missing Network Segmentation: 2 / 2 Risks

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

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.

### Impact

If this risk is unmitigated, attackers successfully attacking other components of the system might have an easy path towards more valuable targets, as they are not separated by network segmentation.

### Detection Logic

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).

### Risk 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.

### False Positives

When all assets within the network segmentation trust-boundary are hardened and protected to the same extend as if all were containing/processing highly sensitive data.

### Mitigation (Operations): Network Segmentation

Apply a network segmentation trust-boundary around the highly sensitive assets and/or datastores.

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 Network Segmentation** 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

**Missing Network Segmentation** to further encapsulate and protect **Apache Webserver** against unrelated lower protected assets in the same network segment, which might be easier to compromise by attackers: Exploitation likelihood is *Unlikely* with *Low* impact.

[missing-network-segmentation@apache-webserver](#)

**Unchecked**

**Missing Network Segmentation** to further encapsulate and protect **Jenkins Buildserver** against unrelated lower protected assets in the same network segment, which might be easier to compromise by attackers: Exploitation likelihood is *Unlikely* with *Low* impact.

[missing-network-segmentation@jenkins-buildserver](#)

**Unchecked**

## LDAP-Injection: 0 / 2 Risks

**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 **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

**LDAP-Injection** risk at **Identity Provider** against LDAP server **LDAP Auth Server** via **LDAP Credential Check Traffic**: Exploitation likelihood is *Likely* with *High* impact.

[ldap-injection@identity-provider@ldap-auth-server@identity-provider>ldap-credential-check-traffic](#)

**Mitigated**      2020-01-05    John Doe      XYZ-5678  
The hardening measures were implemented and checked

**LDAP-Injection** risk at **Marketing CMS** against LDAP server **LDAP Auth Server** via **Auth Traffic**: Exploitation likelihood is *Likely* with *High* impact.

[ldap-injection@marketing-cms@ldap-auth-server@marketing-cms>auth-traffic](#)

**Mitigated**      2020-01-05    John Doe      XYZ-5678  
The hardening measures were implemented and checked

## Missing Hardening: 0 / 6 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 **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.

### 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 Backoffice ERP System:** Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@erp-system](#)

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

**Missing Hardening risk at Customer Contract Database:** Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@sql-database](#)

**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 Hardening risk at Jenkins Buildserver:** Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@jenkins-buildserver](#)

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

**Missing Hardening risk at LDAP Auth Server:** Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@ldap-auth-server](#)

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



## Missing Two-Factor Authentication (2FA): 0 / 9 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 **9 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 **CMS Content Traffic** from **Customer Web Client** forwarded via **Load Balancer** to **Marketing CMS**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@load-balancer>cms-content-traffic@load-balancer@marketing-cms

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

**Missing Two-Factor Authentication** covering communication link **DB Update Access** from **Backend Admin Client** to **Customer Contract Database**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>db-update-access@backend-admin-client@sql-database

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

**Missing Two-Factor Authentication** covering communication link **ERP Internal Access** from **Backoffice Client** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backoffice-client>erp-internal-access@backoffice-client@erp-system

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

**Missing Two-Factor Authentication** covering communication link **ERP Web Access** from **Backend Admin Client** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>erp-web-access@backend-admin-client@erp-system

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

**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 **Jenkins Web-UI Access** from **External Development Client** to **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@external-dev-client>jenkins-web-ui-access@external-dev-client@jenkins-buildserver

**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 **LDAP Auth Server**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>user-management-access@backend-admin-client@ldap-auth-server

**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

## Unencrypted Technical Assets: 0 / 8 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 **8 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](#)

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

**Unencrypted Technical Asset** named **Backoffice ERP System** missing enduser-individual encryption with data-with-enduser-individual-key: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@erp-system](#)

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

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

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

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

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

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

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

**Unencrypted Technical Asset** named **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@jenkins-buildserver](#)

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

**Unencrypted Technical Asset** named **Marketing CMS**: Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@marketing-cms](#)

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

**Unencrypted Technical Asset** named **Contract Fileserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@contract-fileserver

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

**Unencrypted Technical Asset** named **Customer Contract Database** missing enduser-individual encryption with data-with-enduser-individual-key: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@sql-database

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

## Identified Risks by Technical Asset

In total **84 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 2 as high, 29 as elevated, 44 as medium, and 8 as low.**

These risks are distributed across **10 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.

## Customer Contract Database: 1 / 4 Risks

### Description

The database behind the ERP system

### 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@sql-database

Unchecked

#### Elevated Risk Severity

**Missing Hardening** risk at **Customer Contract Database:** Exploitation likelihood is *Likely* with *Medium* impact.

missing-hardening@sql-database

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

#### Medium Risk Severity

**Missing Two-Factor Authentication** covering communication link **DB Update Access** from **Backend Admin Client** to **Customer Contract Database:** Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>db-update-access@backend-admin-client@sql-database

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

**Unencrypted Technical Asset** named **Customer Contract Database** missing enduser-individual encryption with data-with-enduser-individual-key: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@sql-database

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

### Asset Information

ID: sql-database  
Type: datastore



Usage:	business
RAA:	96 %
Size:	component
Technology:	database
Tags:	linux, mysql
Internet:	false
Machine:	virtual
Encryption:	data-with-symmetric-shared-key
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Database Customizing and Dumps
Data Stored:	Customer Accounts, Customer Operational Data, Some Internal Business Data
Formats Accepted:	none of the special data formats accepted

## Asset Rating

Owner:	Company ABC
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 ERP system's database contains business-relevant sensitive data for the leasing processes and eventually also for other Company XYZ internal processes.

## Incoming Communication Links: 2

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

### Database Traffic (incoming)

Link to the DB system

Source:	Backoffice ERP System
Protocol:	jdbc
Encrypted:	false
Authentication:	credentials
Authorization:	technical-user

Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts, Customer Operational Data, Some Internal Business Data
Data Sent:	Customer Accounts, Customer Operational Data, Some Internal Business Data

### DB Update Access (incoming)

Link to the database (JDBC tunneled via SSH)

Source:	Backend Admin Client
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Database Customizing and Dumps
Data Sent:	Customer Accounts, Customer Operational Data, Database Customizing and Dumps, ERP Logs

## Backoffice ERP System: 15 / 19 Risks

### Description

ERP system

### Identified Risks of Asset

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

#### High Risk Severity

**SQL/NoSQL-Injection** risk at **Backoffice ERP System** against database **Customer Contract Database** via **Database Traffic**: Exploitation likelihood is *Very Likely* with *High* impact.

sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic

Unchecked

**XML External Entity (XXE)** risk at **Backoffice ERP System**: Exploitation likelihood is *Very Likely* with *High* impact.

xml-external-entity@erp-system

Unchecked

#### Elevated Risk Severity

**Cross-Site Scripting (XSS)** risk at **Backoffice ERP System**: Exploitation likelihood is *Likely* with *High* impact.

cross-site-scripting@erp-system

Unchecked

**Path-Traversal** risk at **Backoffice ERP System** against filesystem **Contract Fileserver** via **NFS Filesystem Access**: Exploitation likelihood is *Very Likely* with *Medium* impact.

path-traversal@erp-system@contract-fileserver@erp-system>nfs-filesystem-access

Unchecked

**Untrusted Deserialization** risk at **Backoffice ERP System**: Exploitation likelihood is *Likely* with *Very High* impact.

untrusted-deserialization@erp-system

Accepted	2020-01-04	John Doe	XYZ-1234
Risk accepted as tolerable			

**Missing Hardening** risk at **Backoffice ERP System**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-hardening@erp-system

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

## Medium Risk Severity

**Unencrypted Communication** named **Database Traffic** between **Backoffice ERP System** and **Customer Contract Database** transferring authentication data (like credentials, token, session-id, etc.): Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-communication@erp-system>database-traffic@erp-system@sql-database

**Unchecked**

**Missing Enduser Identity Propagation** over communication link **ERP System Traffic** from **Apache Webserver** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

**Unchecked**

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

missing-vault@erp-system

**Unchecked**

**Missing Web Application Firewall (WAF)** risk at **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-waf@erp-system

**Unchecked**

**Unencrypted Communication** named **NFS Filesystem Access** between **Backoffice ERP System** and **Contract Fileserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-communication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver

**Unchecked**

**Cross-Site Request Forgery (CSRF)** risk at **Backoffice ERP System** via **ERP Internal Access** from **Backoffice Client**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@erp-system@backoffice-client>erp-internal-access

**Unchecked**

**Cross-Site Request Forgery (CSRF)** risk at **Backoffice ERP System** via **ERP System Traffic** from **Apache Webserver**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

**Unchecked**

**Cross-Site Request Forgery (CSRF)** risk at **Backoffice ERP System** via **ERP Web Access** from **Backend Admin Client**: Exploitation likelihood is *Likely* with *Low* impact.

cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

**Unchecked**

**Unencrypted Technical Asset** named **Backoffice ERP System** missing enduser-individual encryption with data-with-enduser-individual-key: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@erp-system

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

**Missing Two-Factor Authentication** covering communication link **ERP Internal Access** from **Backoffice Client** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backoffice-client>erp-internal-access@backoffice-client@erp-system

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

**Missing Two-Factor Authentication** covering communication link **ERP Web Access** from **Backend Admin Client** to **Backoffice ERP System**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@backend-admin-client>erp-web-access@backend-admin-client@erp-system

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

## Low Risk Severity

**Denial-of-Service** risky access of **Backoffice ERP System** by **Apache Webserver** via **ERP System Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

dos-risky-access-across-trust-boundary@erp-system@apache-webserver@apache-webserver>erp-system-traffic

**in Progress** 2020-01-04 John Doe XYZ-1234  
The hardening measures are being implemented and checked

**Denial-of-Service** risky access of **Backoffice ERP System** by **Backoffice Client** via **ERP Internal Access**: Exploitation likelihood is *Unlikely* with *Low* impact.

dos-risky-access-across-trust-boundary@erp-system@backoffice-client@backoffice-client>erp-internal-access

**in Progress** 2020-01-04 John Doe XYZ-1234  
The hardening measures are being implemented and checked

## Asset Information

ID:	erp-system
Type:	process
Usage:	business
RAA:	100 %
Size:	system
Technology:	erp
Tags:	linux
Internet:	false
Machine:	virtual

Encryption:	none
Multi-Tenant:	false
Redundant:	true
Custom-Developed:	false
Client by Human:	false
Data Processed:	Customer Accounts, Customer Contracts, Customer Operational Data, ERP Customizing Data, Some Internal Business Data
Data Stored:	ERP Logs
Formats Accepted:	File, Serialization, XML

## Asset Rating

Owner:	Company ABC
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 ERP system contains business-relevant sensitive data for the leasing processes and eventually also for other Company XYZ internal processes.

## Outgoing Communication Links: 2

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

### NFS Filesystem Access (outgoing)

Link to the file system

Target:	Contract Fileserver
Protocol:	nfs
Encrypted:	false
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Contracts
Data Received:	Customer Contracts

**Database Traffic (outgoing)**

Link to the DB system

Target:	Customer Contract Database
Protocol:	jdbc
Encrypted:	false
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Accounts, Customer Operational Data, Some Internal Business Data
Data Received:	Customer Accounts, Customer Operational Data, Some Internal Business Data

**Incoming Communication Links: 3**

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

**ERP Internal Access (incoming)**

Link to the ERP system

Source:	Backoffice Client
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	some-erp
VPN:	true
IP-Filtered:	false
Data Received:	Some Internal Business Data
Data Sent:	Customer Contracts, Some Internal Business Data

**ERP Web Access (incoming)**

Link to the ERP system (Web)

Source:	Backend Admin Client
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	ERP Customizing Data
Data Sent:	ERP Logs

#### ERP System Traffic (incoming)

Link to the ERP system

Source:	Apache Webserver
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts, Customer Operational Data, Some Internal Business Data
Data Sent:	Customer Accounts, Customer Contracts, Customer Operational Data, Some Internal Business Data



## Apache Webserver: 11 / 14 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

**Missing Cloud Hardening (EC2)** risk at **Apache Webserver**: [CIS Benchmark for Amazon Linux](#): Exploitation likelihood is *Unlikely* with *Very High* impact.

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

**Unchecked**

**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 **Backoffice ERP System** via **ERP System Traffic**: Exploitation likelihood is *Likely* with *Medium* impact.

[server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic](#)

**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

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

missing-waf@apache-webserver

Unchecked

**Push instead of Pull Deployment** at **Apache Webserver** via build pipeline asset **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

push-instead-of-pull-deployment@jenkins-buildserver

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

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

unencrypted-asset@apache-webserver

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

## Low Risk Severity

**Missing Network Segmentation** to further encapsulate and protect **Apache Webserver** against unrelated lower protected assets in the same network segment, which might be easier to compromise by attackers: Exploitation likelihood is *Unlikely* with *Low* impact.

missing-network-segmentation@apache-webserver

Unchecked

**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:	98 %
Size:	application
Technology:	web-server
Tags:	apache, aws:ec2, linux
Internet:	false
Machine:	container
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	true
Client by Human:	false
Data Processed:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data, Server Application Code, Some Internal Business Data
Data Stored:	Client 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: 2

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

### ERP System Traffic (outgoing)

[Link to the ERP system](#)

Target:	Backoffice ERP System
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	technical-user
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Accounts, Customer Operational Data, Some Internal Business Data
Data Received:	Customer Accounts, Customer Contracts, Customer Operational Data, Some Internal Business Data

### 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: 2

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, Customer Operational Data
Data Sent:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data

### Application Deployment (incoming)

[Link to the Apache webserver](#)

Source:	Jenkins Buildserver
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Client Application Code, Server Application Code
Data Sent:	none

## Contract Fileserver: 3 / 4 Risks

### Description

NFS Filesystem for storing the contract PDFs

### Identified Risks of Asset

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

#### *Elevated Risk Severity*

**Missing Authentication** covering communication link **NFS Filesystem Access** from **Backoffice ERP System to Contract Fileserver**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-authentication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver

**Unchecked**

#### *Medium Risk Severity*

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

something-strange@contract-fileserver

**Unchecked**

**Missing Cloud Hardening (S3)** risk at **Contract Fileserver**: Security Best Practices for AWS S3: Exploitation likelihood is *Unlikely* with *High* impact.

missing-cloud-hardening@contract-fileserver

**Unchecked**

**Unencrypted Technical Asset** named **Contract Fileserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@contract-fileserver

**Mitigated**

2020-01-04 John Doe

XYZ-1234

The hardening measures were implemented and checked

### Asset Information

ID:	contract-fileserver
Type:	datastore
Usage:	business
RAA:	34 %
Size:	component
Technology:	file-server

Tags:	aws:s3, linux
Internet:	false
Machine:	virtual
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	none
Data Stored:	Customer Contract Summaries, Customer Contracts
Formats Accepted:	File

## Asset Rating

Owner:	Company ABC	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	critical	(rated 4 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	Contract data might contain financial data as well as personally identifiable information (PII). The integrity and availability of contract data is required for clearing payment disputes. The filesystem is also required to be available for storing new contracts of freshly generated customers.	

## Incoming Communication Links: 1

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

### NFS Filesystem Access (incoming)

Link to the file system

Source:	Backoffice ERP System
Protocol:	nfs
Encrypted:	false
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false

IP-Filtered: false  
Data Received: Customer Contracts  
Data Sent: Customer Contracts



## Git Repository: 5 / 8 Risks

### Description

Git repository server

### Identified Risks of Asset

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

#### 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 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

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

**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:	48 %
Size:	system
Technology:	sourcecode-repository
Tags:	git, linux
Internet:	false
Machine:	virtual
Encryption:	none
Multi-Tenant:	true
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Client Application Code, Server Application Code
Data Stored:	Client Application Code, Server Application Code
Formats Accepted:	File

## Asset Rating

Owner:	Company ABC	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)

CIA-Justification: The code repo pipeline might contain sensitive configuration values like backend credentials, certificates etc. and is therefore rated as confidential.

### Incoming Communication Links: 3

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

#### Git Repo Code Read Access (incoming)

[Link to the Git repository server](#)

Source:	Jenkins Buildserver
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	true
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	none
Data Sent:	Client Application Code, Server Application Code

#### Git-Repo Web-UI Access (incoming)

[Link to the Git repo](#)

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

## Identity Provider: 4 / 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**

**LDAP-Injection** risk at **Identity Provider** against LDAP server **LDAP Auth Server** via **LDAP Credential Check Traffic**: Exploitation likelihood is *Likely* with *High* impact.

[ldap-injection@identity-provider@ldap-auth-server@identity-provider>ldap-credential-check-traffic](#)

**Mitigated**

2020-01-05

John Doe

XYZ-5678

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

#### *Medium Risk Severity*

**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**

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

[unencrypted-asset@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:	65 %
Size:	component
Technology:	identity-provider
Tags:	jboss, keycloak, linux
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:	none of the special data formats accepted

## Asset Rating

Owner:	Company ABC	
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	

## Outgoing Communication Links: 1

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

**LDAP Credential Check Traffic (outgoing)**

Link to the LDAP server

Target:	LDAP Auth Server
Protocol:	ldaps
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.

**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

## Jenkins Buildserver: 5 / 8 Risks

### Description

Jenkins buildserver

### Identified Risks of Asset

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

#### Elevated Risk Severity

**Untrusted Deserialization risk at Jenkins Buildserver:** Exploitation likelihood is *Likely* with *Very High* impact.

[untrusted-deserialization@jenkins-buildserver](#)

Unchecked

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

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

Unchecked

**Missing Hardening risk at Jenkins Buildserver:** Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@jenkins-buildserver](#)

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

#### Medium Risk Severity

**Code Backdooring risk at Jenkins Buildserver:** Exploitation likelihood is *Unlikely* with *High* impact.

[code-backdooring@jenkins-buildserver](#)

Unchecked

**Unchecked Deployment risk at Jenkins Buildserver:** Exploitation likelihood is *Unlikely* with *Medium* impact.

[unchecked-deployment@jenkins-buildserver](#)

Unchecked

**Unencrypted Technical Asset named Jenkins Buildserver:** Exploitation likelihood is *Unlikely* with *High* impact.

[unencrypted-asset@jenkins-buildserver](#)

Mitigated

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked



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

missing-authentication-second-factor@external-dev-client>jenkins-web-ui-access@external-dev-client@jenkins-buildserver

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

### Low Risk Severity

**Missing Network Segmentation** to further encapsulate and protect **Jenkins Buildserver** against unrelated lower protected assets in the same network segment, which might be easier to compromise by attackers: Exploitation likelihood is *Unlikely* with *Low* impact.

missing-network-segmentation@jenkins-buildserver

**Unchecked**

### Asset Information

ID:	jenkins-buildserver
Type:	process
Usage:	devops
RAA:	99 %
Size:	system
Technology:	build-pipeline
Tags:	jenkins, linux
Internet:	false
Machine:	virtual
Encryption:	none
Multi-Tenant:	true
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Build Job Config, Client Application Code, Marketing Material, Server Application Code
Data Stored:	Build Job Config, Client Application Code, Marketing Material, Server Application Code
Formats Accepted:	File, Serialization

### Asset Rating

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

Integrity:	critical	(rated 4 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	The build pipeline might contain sensitive configuration values like backend credentials, certificates etc. and is therefore rated as confidential. The integrity and availability is rated as critical and important due to the risk of reputation damage and application update unavailability when the build pipeline is compromised.	

### Outgoing Communication Links: 3

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

#### Git Repo Code Read Access (outgoing)

Link to the Git repository server

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

#### CMS Updates (outgoing)

Link to the CMS

Target:	Marketing CMS
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false

IP-Filtered: false  
Data Sent: Marketing Material  
Data Received: none

### Application Deployment (outgoing)

[Link to the Apache webserver](#)

Target: Apache Webserver  
Protocol: ssh  
Encrypted: true  
Authentication: client-certificate  
Authorization: technical-user  
Read-Only: false  
Usage: devops  
Tags: none  
VPN: false  
IP-Filtered: false  
Data Sent: Client Application Code, Server Application Code  
Data Received: none

### Incoming Communication Links: 1

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

### Jenkins Web-UI Access (incoming)

[Link to the Jenkins build server](#)

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: Build Job Config  
Data Sent: Build Job Config

## Load Balancer: 1 / 1 Risk

### Description

Load Balancer (HA-Proxy)

### 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**

### Asset Information

ID:	load-balancer
Type:	process
Usage:	business
RAA:	17 %
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:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data, Marketing Material, Some Internal Business Data
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: 2

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, Customer Operational Data
Data Received:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data

### CMS Content Traffic (outgoing)

Link to the CMS server

Target:	Marketing CMS
Protocol:	http
Encrypted:	false
Authentication:	none
Authorization:	none
Read-Only:	true
Usage:	business

Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	none
Data Received:	Marketing Material

### 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:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Customer Accounts, Customer Operational Data
Data Sent:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data, Marketing Material

## Marketing CMS: 8 / 11 Risks

### Description

CMS for the marketing content

### Identified Risks of Asset

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

#### Elevated Risk Severity

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

[cross-site-scripting@marketing-cms](#)

Unchecked

**Unencrypted Communication** named **Auth Traffic** between **Marketing CMS** and **LDAP Auth Server** transferring authentication data (like credentials, token, session-id, etc.): Exploitation likelihood is *Likely* with *High* impact.

[unencrypted-communication@marketing-cms>auth-traffic@marketing-cms@ldap-auth-server](#)

Unchecked

**Missing Authentication** covering communication link **CMS Content Traffic** from **Load Balancer** to **Marketing CMS**: Exploitation likelihood is *Likely* with *Medium* impact.

[missing-authentication@load-balancer>cms-content-traffic@load-balancer@marketing-cms](#)

Unchecked

**LDAP-Injection** risk at **Marketing CMS** against LDAP server **LDAP Auth Server** via **Auth Traffic**: Exploitation likelihood is *Likely* with *High* impact.

[ldap-injection@marketing-cms@ldap-auth-server@marketing-cms>auth-traffic](#)

Mitigated

2020-01-05

John Doe

XYZ-5678

The hardening measures were implemented and checked

#### Medium Risk Severity

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

[container-baseimage-backdooring@marketing-cms](#)

Unchecked

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

[missing-waf@marketing-cms](#)

Unchecked

**Push instead of Pull Deployment at Marketing CMS** via build pipeline asset **Jenkins Buildserver**: Exploitation likelihood is *Unlikely* with *Medium* impact.

push-instead-of-pull-deployment@jenkins-buildserver

**Unchecked****Cross-Site Request Forgery (CSRF)** risk at **Marketing CMS** via **CMS Content Traffic** from **Load Balancer**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@marketing-cms@load-balancer&gt;cms-content-traffic

**Unchecked****Cross-Site Request Forgery (CSRF)** risk at **Marketing CMS** via **Marketing CMS Editing** from **Backoffice Client**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@marketing-cms@backoffice-client&gt;marketing-cms-editing

**Unchecked****Unencrypted Technical Asset** named **Marketing CMS**: Exploitation likelihood is *Unlikely* with *High* impact.

unencrypted-asset@marketing-cms

**Mitigated**

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

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

missing-authentication-second-factor@load-balancer&gt;cms-content-traffic@load-balancer@marketing-cms

**Mitigated**

2020-01-04

John Doe

XYZ-1234

The hardening measures were implemented and checked

**Asset Information**

ID:	marketing-cms
Type:	process
Usage:	business
RAA:	34 %
Size:	application
Technology:	cms
Tags:	linux
Internet:	false
Machine:	container
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	true
Client by Human:	false
Data Processed:	Customer Accounts, Marketing Material



Data Stored: Marketing Material  
Formats Accepted: none of the special data formats accepted

## Asset Rating

Owner: Company ABC  
Confidentiality: internal (rated 2 in scale of 5)  
Integrity: important (rated 3 in scale of 5)  
Availability: important (rated 3 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 Traffic (outgoing)

Link to the LDAP auth server

Target:	LDAP Auth Server
Protocol:	ldap
Encrypted:	false
Authentication:	credentials
Authorization:	technical-user
Read-Only:	true
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Accounts
Data Received:	Customer Accounts

## Incoming Communication Links: 3

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

### CMS Content Traffic (incoming)

Link to the CMS server

Source:	Load Balancer
---------	---------------

Protocol:	http
Encrypted:	false
Authentication:	none
Authorization:	none
Read-Only:	true
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	none
Data Sent:	Marketing Material

### CMS Updates (incoming)

Link to the CMS

Source:	Jenkins Buildserver
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Marketing Material
Data Sent:	none

### Marketing CMS Editing (incoming)

Link to the CMS for editing content

Source:	Backoffice Client
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true

IP-Filtered: false  
Data Received: Marketing Material  
Data Sent: Marketing Material

## LDAP Auth Server: 1 / 3 Risks

### Description

LDAP authentication server

### Identified Risks of Asset

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

#### Elevated Risk Severity

**Missing Hardening** risk at **LDAP Auth Server**: Exploitation likelihood is *Likely* with *Medium* impact.

[missing-hardening@ldap-auth-server](#)

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

#### Medium Risk Severity

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

[missing-authentication-second-factor@backend-admin-client>user-management-access@backend-admin-client@ldap-auth-server](#)

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

#### Low Risk Severity

**Denial-of-Service** risky access of **LDAP Auth Server** by **Marketing CMS** via **Auth Traffic**: Exploitation likelihood is *Unlikely* with *Low* impact.

[dos-risky-access-across-trust-boundary@ldap-auth-server@marketing-cms@marketing-cms>auth-traffic](#)

**in Progress** 2020-01-04 John Doe XYZ-1234  
The hardening measures are being implemented and checked

### Asset Information

ID:	ldap-auth-server
Type:	datastore
Usage:	business
RAA:	83 %
Size:	component
Technology:	identity-store-ldap

Tags:	linux
Internet:	false
Machine:	physical
Encryption:	transparent
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Customer Accounts
Data Stored:	Customer Accounts
Formats Accepted:	none of the special data formats accepted

## Asset Rating

Owner:	Company ABC	
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: 3

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

### Auth Traffic (incoming)

Link to the LDAP auth server

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

**LDAP Credential Check Traffic (incoming)**

Link to the LDAP server

Source:	Identity Provider
Protocol:	ldaps
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

**User Management Access (incoming)**

Link to the LDAP auth server for managing users

Source:	Backend Admin Client
Protocol:	ldaps
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

# Backend Admin Client: out-of-scope

## Description

Backend admin client

## Identified Risks of Asset

Asset was defined as out-of-scope.

## Asset Information

ID:	backend-admin-client
Type:	external-entity
Usage:	devops
RAA:	out-of-scope
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:	ERP Logs
Data Stored:	none
Formats Accepted:	none of the special data formats accepted

## Asset Rating

Owner:	Company XYZ	
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 Company XYZ to administer the system.	

## Asset Out-of-Scope Justification

Owned and managed by ops provider

### Outgoing Communication Links: 3

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

#### User Management Access (outgoing)

Link to the LDAP auth server for managing users

Target:	LDAP Auth Server
Protocol:	ldaps
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

#### ERP Web Access (outgoing)

Link to the ERP system (Web)

Target:	Backoffice ERP System
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	ERP Customizing Data
Data Received:	ERP Logs



## DB Update Access (outgoing)

Link to the database (JDBC tunneled via SSH)

Target:	Customer Contract Database
Protocol:	ssh
Encrypted:	true
Authentication:	client-certificate
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Database Customizing and Dumps
Data Received:	Customer Accounts, Customer Operational Data, Database Customizing and Dumps, ERP Logs

## Backoffice Client: out-of-scope

### Description

Backoffice client

### Identified Risks of Asset

Asset was defined as out-of-scope.

### Asset Information

ID:	backoffice-client
Type:	external-entity
Usage:	business
RAA:	out-of-scope
Size:	component
Technology:	desktop
Tags:	none
Internet:	false
Machine:	physical
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	true
Data Processed:	Customer Contracts, ERP Logs, Some Internal Business Data
Data Stored:	none
Formats Accepted:	none of the special data formats accepted

### Asset Rating

Owner:	Company XYZ	
Confidentiality:	confidential	(rated 4 in scale of 5)
Integrity:	important	(rated 3 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	The client used by Company XYZ to administer and use the system.	

## Asset Out-of-Scope Justification

Owned and managed by Company XYZ company

## Outgoing Communication Links: 2

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

### Marketing CMS Editing (outgoing)

Link to the CMS for editing content

Target:	Marketing CMS
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	true
IP-Filtered:	false
Data Sent:	Marketing Material
Data Received:	Marketing Material

### ERP Internal Access (outgoing)

Link to the ERP system

Target:	Backoffice ERP System
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	some-erp
VPN:	true
IP-Filtered:	false
Data Sent:	Some Internal Business Data
Data Received:	Customer Contracts, Some Internal Business Data

## 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:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data, 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:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Customer Accounts, Customer Operational Data
Data Received:	Client Application Code, Customer Accounts, Customer Contracts, Customer Operational Data, 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:	linux
Internet:	true
Machine:	physical
Encryption:	none
Multi-Tenant:	true
Redundant:	false
Custom-Developed:	false
Client by Human:	true
Data Processed:	Client Application Code, Server Application Code
Data Stored:	Client 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: 3

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

### Jenkins Web-UI Access (outgoing)

Link to the Jenkins build server

Target:	Jenkins Buildserver
Protocol:	https
Encrypted:	true
Authentication:	credentials
Authorization:	technical-user
Read-Only:	false
Usage:	devops
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Build Job Config
Data Received:	Build Job Config

### Git-Repo Web-UI Access (outgoing)

Link to the Git repo

Target:	Git Repository
Protocol:	https
Encrypted:	true
Authentication:	token
Authorization:	technical-user
Read-Only:	false

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

#### Git-Repo Code Write Access (outgoing)

Link to the Git repo

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



## Identified Data Breach Probabilities by Data Asset

In total **84 potential risks** have been identified during the threat modeling process of which **1 are rated as critical, 2 as high, 29 as elevated, 44 as medium, and 8 as low.**

These risks are distributed across **12 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.

## Build Job Config: 6 / 9 Risks

Data for customizing of the build job system.

ID:	build-job-config
Usage:	devops
Quantity:	very-few
Tags:	none
Origin:	Company XYZ
Owner:	Company XYZ
Confidentiality:	restricted (rated 3 in scale of 5)
Integrity:	critical (rated 4 in scale of 5)
Availability:	operational (rated 2 in scale of 5)
CIA-Justification:	Data for customizing of the build job system.
Processed by:	Jenkins Buildserver
Stored by:	Jenkins Buildserver
Sent via:	Jenkins Web-UI Access
Received via:	Jenkins Web-UI Access
Data Breach:	<b>probable</b>

Data Breach Risks: This data asset has data breach potential because of 6 remaining risks:

Probable: `code-backdooring@jenkins-buildserver`

Probable: `untrusted-deserialization@jenkins-buildserver`

Possible: `unchecked-deployment@external-dev-client`

Possible: `unchecked-deployment@jenkins-buildserver`

Possible: `unguarded-access-from-internet@jenkins-buildserver@external-dev-client@external-dev-client>jenkins-web-ui-access`

Improbable: `missing-network-segmentation@jenkins-buildserver`

## Client Application Code: 25 / 34 Risks

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

ID:	client-application-code	
Usage:	devops	
Quantity:	very-few	
Tags:	none	
Origin:	Company ABC	
Owner:	Company ABC	
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, Customer Web Client, External Development Client, Git Repository, Jenkins Buildserver, Load Balancer	
Stored by:	Apache Webserver, External Development Client, Git Repository, Jenkins Buildserver	
Sent via:	Git-Repo Web-UI Access, Git-Repo Code Write Access, Application Deployment	
Received via:	Web Application Traffic, Git-Repo Web-UI Access, Git-Repo Code Write Access, Git Repo Code Read Access, Customer Traffic	
Data Breach:	<b>probable</b>	
Data Breach Risks:	This data asset has data breach potential because of 25 remaining risks:	
	Probable: accidental-secret-leak@git-repo	
	Probable: code-backdooring@git-repo	
	Probable: code-backdooring@jenkins-buildserver	
	Probable: container-baseimage-backdooring@apache-webserver	
	Probable: missing-cloud-hardening@application-network	
	Probable: missing-cloud-hardening@apache-webserver	
	Probable: missing-cloud-hardening@web-dmz	
	Probable: missing-file-validation@apache-webserver	
	Probable: untrusted-deserialization@jenkins-buildserver	
	Possible: cross-site-scripting@apache-webserver	
	Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic	
	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: unchecked-deployment@jenkins-buildserver	
	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`

Possible: `unguarded-access-from-internet@jenkins-buildserver@external-dev-client@external-dev-client>jenkins-web-ui-access`

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

Improbable: `missing-network-segmentation@apache-webserver`

Improbable: `missing-network-segmentation@jenkins-buildserver`

Improbable: `missing-waf@apache-webserver`

Improbable: `mixed-targets-on-shared-runtime@webapp-virtualization`

Improbable: `push-instead-of-pull-deployment@jenkins-buildserver`

## Customer Accounts: 39 / 57 Risks

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

ID:	customer-accounts
Usage:	business
Quantity:	many
Tags:	none
Origin:	Customer
Owner:	Company XYZ
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 account data for using the portal are required to be available to offer the portal functionality.
Processed by:	Apache Webserver, Backoffice ERP System, Customer Web Client, Identity Provider, LDAP Auth Server, Load Balancer, Marketing CMS
Stored by:	Customer Contract Database, LDAP Auth Server
Sent via:	Web Application Traffic, User Management Access, LDAP Credential Check Traffic, ERP System Traffic, Database Traffic, Customer Traffic, Auth Traffic, Auth Credential Check Traffic
Received via:	Web Application Traffic, User Management Access, ERP System Traffic, Database Traffic, DB Update Access, Customer Traffic, Auth Traffic

Data Breach: **probable**

Data Breach Risks: This data asset has data breach potential because of 39 remaining risks:

Probable: code-backdooring@jenkins-buildserver

Probable: container-baseimage-backdooring@apache-webserver

Probable: container-baseimage-backdooring@marketing-cms

Probable: something-strange@sql-database

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@apache-webserver

Probable: missing-cloud-hardening@erp-dmz

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Probable: sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic

Probable: xml-external-entity@erp-system

Probable: untrusted-deserialization@erp-system

Possible: cross-site-scripting@apache-webserver

Possible: cross-site-scripting@erp-system

Possible: cross-site-scripting@identity-provider

Possible: cross-site-scripting@marketing-cms

Possible: missing-authentication@load-balancer>cms-content-traffic@load-balancer@marketing-cms

Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic

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

Possible: unchecked-deployment@jenkins-buildserver

Possible: unencrypted-communication@marketing-cms>auth-traffic@marketing-cms@ldap-auth-server

Possible: unencrypted-communication@erp-system>database-traffic@erp-system@sql-database

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@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

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

Improbable: cross-site-request-forgery@marketing-cms@load-balancer>cms-content-traffic

Improbable: cross-site-request-forgery@marketing-cms@backoffice-client>marketing-cms-editing

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-network-segmentation@apache-webserver

Improbable: missing-waf@apache-webserver

Improbable: missing-waf@erp-system

Improbable: missing-waf@identity-provider

Improbable: missing-waf@marketing-cms

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

## Customer Contract Summaries: 7 / 8 Risks

### Customer Contract Summaries

ID: contract-summaries  
Usage: business  
Quantity: very-few  
Tags: none  
Origin: Customer  
Owner: Company XYZ  
Confidentiality: restricted (rated 3 in scale of 5)  
Integrity: operational (rated 2 in scale of 5)  
Availability: operational (rated 2 in scale of 5)  
CIA-Justification: Just some summaries.  
Processed by: none  
Stored by: Contract Fileserver  
Sent via: none  
Received via: none  
Data Breach: **probable**

Data Breach Risks: This data asset has data breach potential because of 7 remaining risks:

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@contract-fileserver

Probable: missing-cloud-hardening@erp-dmz

Probable: path-traversal@erp-system@contract-fileserver@erp-system>nfs-filesystem-access

Possible: missing-authentication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver

Possible: unencrypted-communication@erp-system>nfs-filesystem-access@erp-system@contract-fileserver

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

## Customer Contracts: 29 / 37 Risks

### Customer Contracts (PDF)

ID:	customer-contracts	
Usage:	business	
Quantity:	many	
Tags:	none	
Origin:	Customer	
Owner:	Company XYZ	
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:	Contract data might contain financial data as well as personally identifiable information (PII). The integrity and availability of contract data is required for clearing payment disputes.	
Processed by:	Apache Webserver, Backoffice Client, Backoffice ERP System, Customer Web Client, Load Balancer	
Stored by:	Contract Fileserver	
Sent via:	NFS Filesystem Access	
Received via:	Web Application Traffic, NFS Filesystem Access, ERP System Traffic, ERP Internal Access, Customer Traffic	

Data Breach: **probable**

Data Breach Risks: This data asset has data breach potential because of 29 remaining risks:

Probable: code-backdooring@jenkins-buildserver

Probable: container-baseimage-backdooring@apache-webserver

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@apache-webserver

Probable: missing-cloud-hardening@contract-fileserver

Probable: missing-cloud-hardening@erp-dmz

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Probable: path-traversal@erp-system@contract-fileserver@erp-system>nfs-file-system-access

Probable: xml-external-entity@erp-system

Probable: untrusted-deserialization@erp-system

Possible: cross-site-scripting@apache-webserver

Possible: cross-site-scripting@erp-system

Possible: missing-authentication@erp-system>nfs-file-system-access@erp-system@contract-fileserver

Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic

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

Possible: unchecked-deployment@jenkins-buildserver

Possible: unencrypted-communication@erp-system>nfs-file-system-access@erp-system@contract-fileserver

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@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-network-segmentation@apache-webserver

Improbable: missing-waf@apache-webserver

Improbable: missing-waf@erp-system

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

## Customer Operational Data: 28 / 38 Risks

### Customer Operational Data

ID:	customer-operational-data	
Usage:	business	
Quantity:	many	
Tags:	none	
Origin:	Customer	
Owner:	Company XYZ	
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:	Customer operational data for using the portal are required to be available to offer the portal functionality and are used in the backend transactions.	
Processed by:	Apache Webserver, Backoffice ERP System, Customer Web Client, Load Balancer	
Stored by:	Customer Contract Database	
Sent via:	Web Application Traffic, ERP System Traffic, Database Traffic, Customer Traffic	
Received via:	Web Application Traffic, ERP System Traffic, Database Traffic, DB Update Access, Customer Traffic	

Data Breach: **probable**

Data Breach Risks: This data asset has data breach potential because of 28 remaining risks:

Probable: code-backdooring@jenkins-buildserver  
 Probable: container-baseimage-backdooring@apache-webserver  
 Probable: something-strange@sql-database  
 Probable: missing-cloud-hardening@application-network  
 Probable: missing-cloud-hardening@apache-webserver  
 Probable: missing-cloud-hardening@erp-dmz  
 Probable: missing-cloud-hardening@web-dmz  
 Probable: missing-file-validation@apache-webserver  
 Probable: sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic  
 Probable: xml-external-entity@erp-system  
 Probable: untrusted-deserialization@erp-system  
 Possible: cross-site-scripting@apache-webserver  
 Possible: cross-site-scripting@erp-system  
 Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic  
 Possible: server-side-request-forgery@apache-webserver@identity-provider@apache-webserver>auth-credential-check-traffic  
 Possible: unchecked-deployment@jenkins-buildserver  
 Possible: unencrypted-communication@erp-system>database-traffic@erp-system@sql-database  
 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@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-network-segmentation@apache-webserver

Improbable: missing-waf@apache-webserver

Improbable: missing-waf@erp-system

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

## Database Customizing and Dumps: 6 / 9 Risks

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

ID: db-dumps  
Usage: devops  
Quantity: very-few  
Tags: oracle  
Origin: Company XYZ  
Owner: Company XYZ  
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: Customer Contract Database  
Stored by: none  
Sent via: DB Update Access  
Received via: DB Update Access  
Data Breach: **probable**  
Data Breach Risks: This data asset has data breach potential because of 6 remaining risks:

Probable: something-strange@sql-database

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@erp-dmz

Probable: sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic

Possible: unencrypted-communication@erp-system>database-traffic@erp-system@sql-database

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

## ERP Customizing Data: 11 / 15 Risks

Data for customizing of the ERP system.

ID:	erp-customizing
Usage:	devops
Quantity:	very-few
Tags:	none
Origin:	Company XYZ
Owner:	Company XYZ
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:	Data for customizing of the ERP system.
Processed by:	Backoffice ERP System
Stored by:	none
Sent via:	ERP Web Access
Received via:	none
Data Breach:	<b>probable</b>

Data Breach Risks: This data asset has data breach potential because of 11 remaining risks:

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@erp-dmz

Probable: xml-external-entity@erp-system

Probable: untrusted-deserialization@erp-system

Possible: cross-site-scripting@erp-system

Improbable: cross-site-request-forgery@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-waf@erp-system

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

## ERP Logs: 11 / 15 Risks

Logs generated by the ERP system.

ID:	erp-logs
Usage:	devops
Quantity:	many
Tags:	none
Origin:	Company XYZ
Owner:	Company XYZ
Confidentiality:	restricted (rated 3 in scale of 5)
Integrity:	archive (rated 1 in scale of 5)
Availability:	archive (rated 1 in scale of 5)
CIA-Justification:	Logs should not contain PII data and are only required for failure analysis, i.e. they are not considered as hard transactional logs.
Processed by:	Backend Admin Client, Backoffice Client
Stored by:	Backoffice ERP System
Sent via:	none
Received via:	ERP Web Access, DB Update Access
Data Breach:	<b>probable</b>
Data Breach Risks:	This data asset has data breach potential because of 11 remaining risks:

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@erp-dmz

Probable: xml-external-entity@erp-system

Probable: untrusted-deserialization@erp-system

Possible: cross-site-scripting@erp-system

Improbable: cross-site-request-forgery@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-waf@erp-system

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

## Marketing Material: 18 / 23 Risks

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

ID:	marketing-material	
Usage:	devops	
Quantity:	very-few	
Tags:	none	
Origin:	Company ABC	
Owner:	Company ABC	
Confidentiality:	public	(rated 1 in scale of 5)
Integrity:	important	(rated 3 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:	Customer Web Client, Jenkins Buildserver, Load Balancer, Marketing CMS	
Stored by:	Jenkins Buildserver, Marketing CMS	
Sent via:	Marketing CMS Editing, CMS Updates	
Received via:	Marketing CMS Editing, Customer Traffic, CMS Content Traffic	
Data Breach:	<b>probable</b>	
Data Breach Risks:	This data asset has data breach potential because of 18 remaining risks:	

Probable: code-backdooring@jenkins-buildserver

Probable: container-baseimage-backdooring@marketing-cms

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@web-dmz

Probable: untrusted-deserialization@jenkins-buildserver

Possible: cross-site-scripting@marketing-cms

Possible: missing-authentication@load-balancer>cms-content-traffic@load-balancer@marketing-cms

Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic

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

Possible: unchecked-deployment@external-dev-client

Possible: unchecked-deployment@jenkins-buildserver

Possible: unguarded-access-from-internet@jenkins-buildserver@external-dev-client@external-dev-client>jenkins-web-ui-access

Improbable: cross-site-request-forgery@marketing-cms@load-balancer>cms-content-traffic

Improbable: cross-site-request-forgery@marketing-cms@backoffice-client>marketing-cms-editing

Improbable: missing-network-segmentation@jenkins-buildserver

Improbable: missing-waf@marketing-cms

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

## Server Application Code: 25 / 34 Risks

API and other server-side code of the application.

ID:	server-application-code	
Usage:	devops	
Quantity:	very-few	
Tags:	none	
Origin:	Company ABC	
Owner:	Company ABC	
Confidentiality:	internal	(rated 2 in scale of 5)
Integrity:	mission-critical	(rated 5 in scale of 5)
Availability:	important	(rated 3 in scale of 5)
CIA-Justification:	The integrity of the API 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, External Development Client, Git Repository, Jenkins Buildserver	
Stored by:	Apache Webserver, External Development Client, Git Repository, Jenkins Buildserver	
Sent via:	Git-Repo Web-UI Access, Git-Repo Code Write Access, Application Deployment	
Received via:	Git-Repo Web-UI Access, Git-Repo Code Write Access, Git Repo Code Read Access	
Data Breach:	<b>probable</b>	
Data Breach Risks:	This data asset has data breach potential because of 25 remaining risks:	

Probable: accidental-secret-leak@git-repo

Probable: code-backdooring@git-repo

Probable: code-backdooring@jenkins-buildserver

Probable: container-baseimage-backdooring@apache-webserver

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@apache-webserver

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Probable: untrusted-deserialization@jenkins-buildserver

Possible: cross-site-scripting@apache-webserver

Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic

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: unchecked-deployment@jenkins-buildserver

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`

Possible: `unguarded-access-from-internet@jenkins-buildserver@external-dev-client@external-dev-client>jenkins-web-ui-access`

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

Improbable: `missing-network-segmentation@apache-webserver`

Improbable: `missing-network-segmentation@jenkins-buildserver`

Improbable: `missing-waf@apache-webserver`

Improbable: `mixed-targets-on-shared-runtime@webapp-virtualization`

Improbable: `push-instead-of-pull-deployment@jenkins-buildserver`

## Some Internal Business Data: 28 / 38 Risks

Internal business data of the ERP system used unrelated to the customer-facing processes.

ID:	internal-business-data
Usage:	business
Quantity:	few
Tags:	none
Origin:	Company XYZ
Owner:	Company XYZ
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 used and/or generated during unrelated other usecases of the ERP-system (when used also by Company XYZ for internal non-customer-portal-related stuff).
Processed by:	Apache Webserver, Backoffice Client, Backoffice ERP System, Load Balancer
Stored by:	Customer Contract Database
Sent via:	ERP System Traffic, ERP Internal Access, Database Traffic
Received via:	ERP System Traffic, ERP Internal Access, Database Traffic
Data Breach:	<b>probable</b>
Data Breach Risks:	This data asset has data breach potential because of 28 remaining risks:

Probable: code-backdooring@jenkins-buildserver

Probable: container-baseimage-backdooring@apache-webserver

Probable: something-strange@sql-database

Probable: missing-cloud-hardening@application-network

Probable: missing-cloud-hardening@apache-webserver

Probable: missing-cloud-hardening@erp-dmz

Probable: missing-cloud-hardening@web-dmz

Probable: missing-file-validation@apache-webserver

Probable: sql-nosql-injection@erp-system@sql-database@erp-system>database-traffic

Probable: xml-external-entity@erp-system

Probable: untrusted-deserialization@erp-system

Possible: cross-site-scripting@apache-webserver

Possible: cross-site-scripting@erp-system

Possible: server-side-request-forgery@apache-webserver@erp-system@apache-webserver>erp-system-traffic

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

Possible: unchecked-deployment@jenkins-buildserver

Possible: unencrypted-communication@erp-system>database-traffic@erp-system@sql-database

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@erp-system@backoffice-client>erp-internal-access

Improbable: cross-site-request-forgery@erp-system@apache-webserver>erp-system-traffic

Improbable: cross-site-request-forgery@erp-system@backend-admin-client>erp-web-access

Improbable: missing-identity-propagation@apache-webserver>erp-system-traffic@apache-webserver@erp-system

Improbable: missing-network-segmentation@apache-webserver

Improbable: missing-waf@apache-webserver

Improbable: missing-waf@erp-system

Improbable: mixed-targets-on-shared-runtime@webapp-virtualization

Improbable: push-instead-of-pull-deployment@jenkins-buildserver

# Trust Boundaries

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

## Application Network

### Application Network

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

## Auth Handling Environment

### Auth Handling Environment

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

## Dev Network

### Development Network

ID: dev-network  
Type: [network-on-prem](#)  
Tags: none  
Assets inside: Backend Admin Client, Backoffice Client, Git Repository, Jenkins  
Buildserver  
Boundaries nested: none

## ERP DMZ

### ERP DMZ

ID: erp-dmz

Type: [network-cloud-security-group](#)  
Tags: some-erp  
Assets inside: Contract Fileserver, Backoffice ERP System, Customer Contract Database  
Boundaries nested: none

## Web DMZ

### Web DMZ

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

# Shared Runtimes

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

## WebApp and Backoffice Virtualization

### WebApp Virtualization

ID:	webapp-virtualization
Tags:	vmware
Assets running:	Apache Webserver, Marketing CMS, Backoffice ERP System, Contract Fileserver, Customer Contract Database

# Risk Rules Checked by Threagile

**Threagile Version:** 1.0.0

**Threagile Build Timestamp:** 20240730113903

**Threagile Execution Timestamp:** 20250104003319

**Model Filename:** /github/workspace/threagile.yaml

**Model Hash (SHA256):** 2714e2f499500d26d7d91fe47357c716d289e8628f64b73fabe5dc0bbf9d777c

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

John Doe 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. John Doe 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 John Doe 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.