Threat Model Report
OWASP Juice Shop — Local Lab Th

18 September 2025

Student Name

Table of Contents

Data Breach Probabilities by Data Asset

Trust Boundaries

Shared Runtime

About Threagile

Management Summary

Threagile toolkit was used to model the architecture of "OWASP Juice Shop — Local Lab Threat Model" 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 "OWASP Juice Shop — Local Lab Threat Model" 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 **20 initial risks** in **14 categories** have been identified during the threat modeling process:

0 critical risk	**20 unchecked**
0 high risk	**0 in discussion**
2 elevated risk	**0 accepted**
13 medium risk	**0 in progress**
5 low risk	**0 mitigated**
	0 false positive
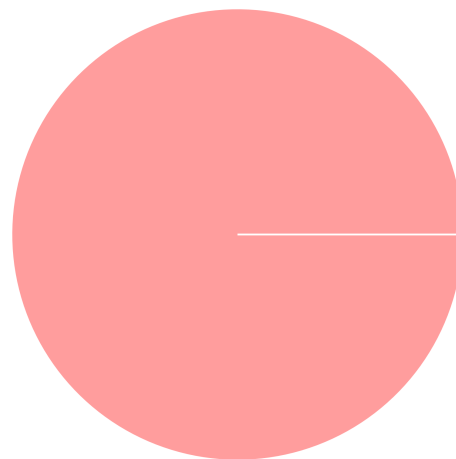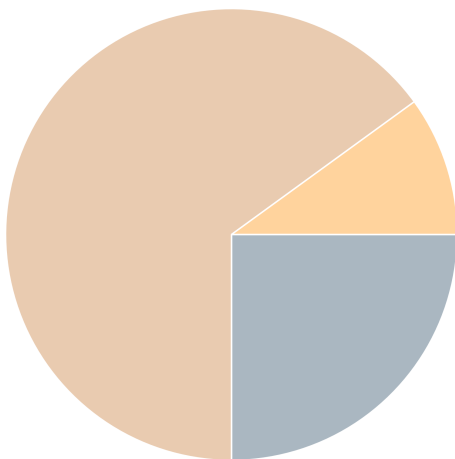
Threat model for a local OWASP Juice Shop setup. Users access the app   either directly via HTTP on port 3000 or through an optional reverse proxy that   terminates TLS and adds security headers. The app runs in a container   and writes data to a host-mounted volume (for database, uploads, logs).   Optional outbound notifications (e.g., a challenge-solution WebHook) can be configured for integrations.

Impact Analysis of 20 Initial Risks in 14 Categories

The most prevalent impacts of the **20 initial risks** (distributed over **14 risk categories**) are (taking the severity ratings into account and using the highest for each category):

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

Elevated: **Cross-Site Scripting (XSS)**: 1 Initial Risk - Exploitation likelihood is *Likely* with *Medium* impact.
If this risk remains unmitigated, attackers might be able to access individual victim sessions and steal or modify user data.

Elevated: **Missing Authentication**: 1 Initial Risk - 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.

Medium: **Container Base Image Backdooring**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to deeply persist in the target system by executing code in deployed containers.

Medium: **Cross-Site Request Forgery (CSRF)**: 2 Initial Risks - Exploitation likelihood is *Very Likely* with *Low* impact.
If this risk remains unmitigated, attackers might be able to trick logged-in victim users into unwanted actions within the web application by visiting an attacker controlled web site.

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

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

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

Medium: **Missing Two-Factor Authentication (2FA)**: 2 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: **Server-Side Request Forgery (SSRF)**: 2 Initial Risks - Exploitation likelihood is *Likely* with *Low* 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.

Medium: **Unencrypted Technical Assets**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to access unencrypted data when successfully compromising sensitive components.

Low: **Missing Web Application Firewall (WAF)**: 1 Initial Risk - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to apply standard attack pattern tests at great speed without any filtering.

Low: **Unnecessary Data Transfer**: 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to target unnecessarily transferred data.

Low: **Unnecessary Technical Asset**: 2 Initial Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to target unnecessary technical assets.

Risk Mitigation

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

20 unchecked
0 in discussion
0 accepted
0 in progress
0 mitigated
0 false positive

Low (5)   Medium (13)   Elevated (2)   High (0)   Critical (0)
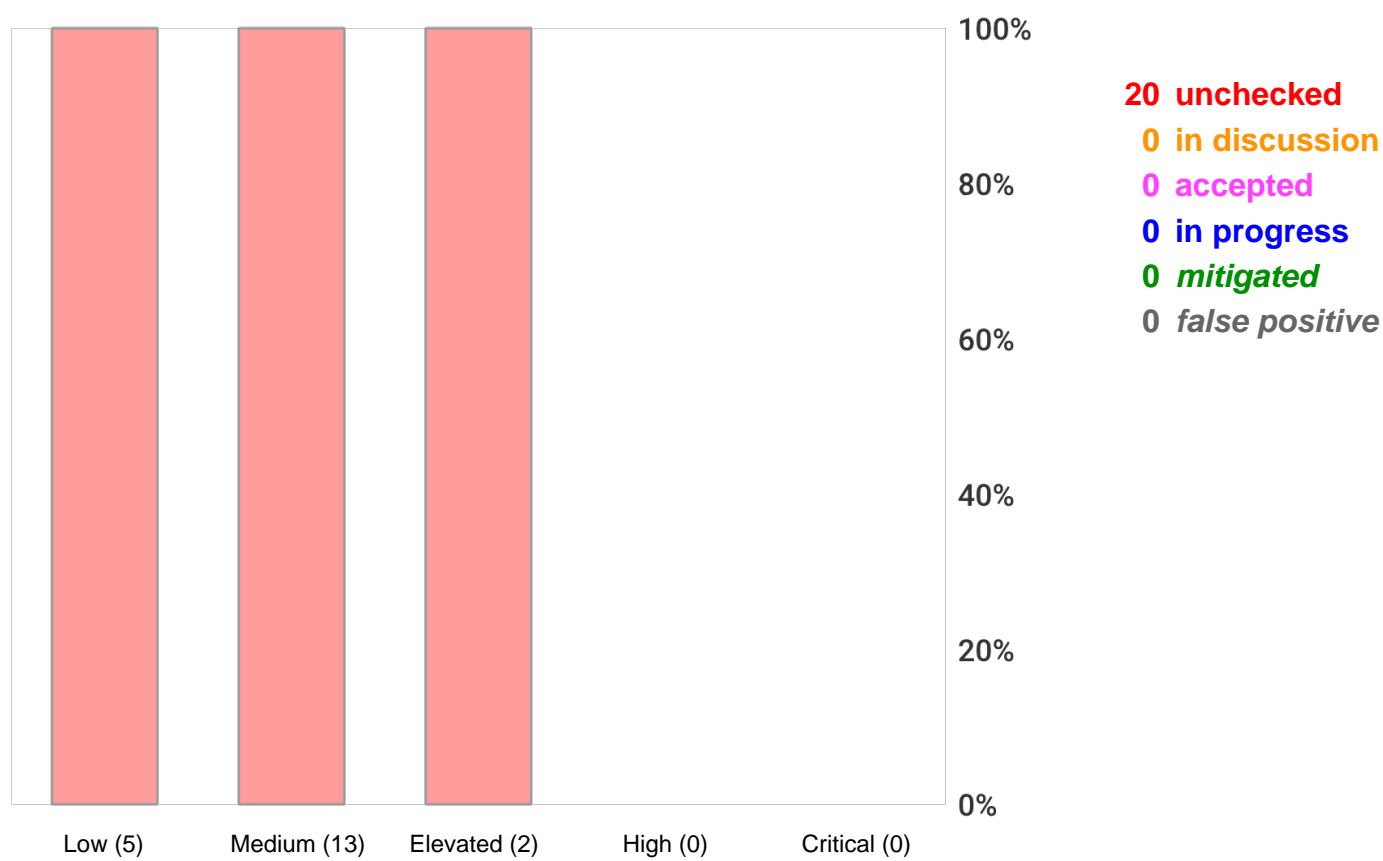
After removal of risks with status *mitigated* and *false positive* the following **20 remain unmitigated**:

0 unmitigated critical risk
0 unmitigated high risk
2 unmitigated elevated risk
13 unmitigated medium risk
5 unmitigated low risk

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

Impact Analysis of 20 Remaining Risks in 14 Categories

The most prevalent impacts of the **20 remaining risks** (distributed over **14 risk categories**) are (taking the severity ratings into account and using the highest for each category):

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

Elevated: **Cross-Site Scripting (XSS)**: 1 Remaining Risk - Exploitation likelihood is *Likely* with *Medium* impact.
If this risk remains unmitigated, attackers might be able to access individual victim sessions and steal or modify user data.

Elevated: **Missing Authentication**: 1 Remaining Risk - 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.

Medium: **Container Base Image Backdooring**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to deeply persist in the target system by executing code in deployed containers.

Medium: **Cross-Site Request Forgery (CSRF)**: 2 Remaining Risks - Exploitation likelihood is *Very Likely* with *Low* impact.
If this risk remains unmitigated, attackers might be able to trick logged-in victim users into unwanted actions within the web application by visiting an attacker controlled web site.

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

Medium: **Missing Hardening**: 2 Remaining Risks - Exploitation likelihood is *Likely* with *Low* impact.
If this risk remains unmitigated, attackers might be able to easier attack high-value targets.
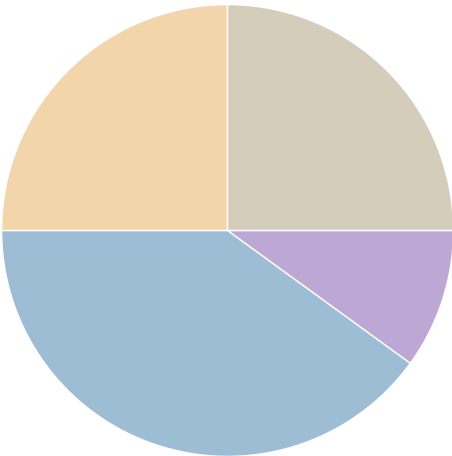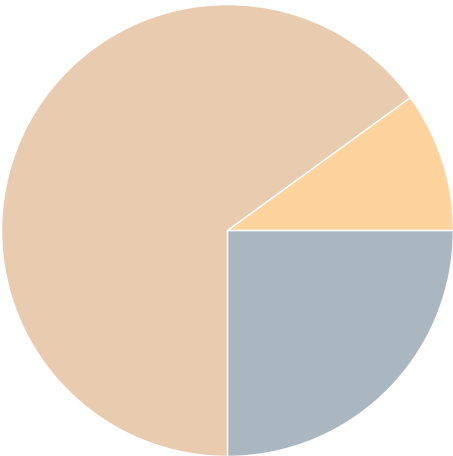
Medium: **Missing Identity Store**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to exploit risks unseen in this threat model in the identity provider/store that is currently missing in the model.

Medium: **Missing Two-Factor Authentication (2FA)**: 2 Remaining 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 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: **Server-Side Request Forgery (SSRF)**: 2 Remaining Risks - Exploitation likelihood is *Likely* with *Low* 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.

Medium: **Unencrypted Technical Assets**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
If this risk is unmitigated, attackers might be able to access unencrypted data when successfully compromising sensitive components.

Low: **Missing Web Application Firewall (WAF)**: 1 Remaining Risk - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to apply standard attack pattern tests at great speed without any filtering.

Low: **Unnecessary Data Transfer**: 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to target unnecessarily transferred data.

Low: **Unnecessary Technical Asset**: 2 Remaining Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
If this risk is unmitigated, attackers might be able to target unnecessary technical assets.

Application Overview

Business Criticality

The overall business criticality of "OWASP Juice Shop — Local Lab Threat Model" was rated as:

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

Business Overview

Training environment for DevSecOps. This model covers a deliberately vulnerable   web application (OWASP Juice Shop) running locally in a Docker container. The focus is on a minimal architecture, STRIDE threat analysis, and actionable mitigations for the identified risks.

Technical Overview

A user's web browser connects to the Juice Shop application (Node.js/Express server) either directly on **localhost:3000** (HTTP) or via a **reverse proxy** on ports 80/443 (with HTTPS). The Juice Shop server may issue outbound requests to external services (e.g., a configured **WebHook** for solved challenge notifications). All application data (the SQLite database, file uploads, logs) is stored on the host's filesystem via a mounted volume. Key trust boundaries include the **Internet** (user & external services) . **Host** (local machine/VM) . **Container Network** (isolated app container).

Data-Flow Diagram

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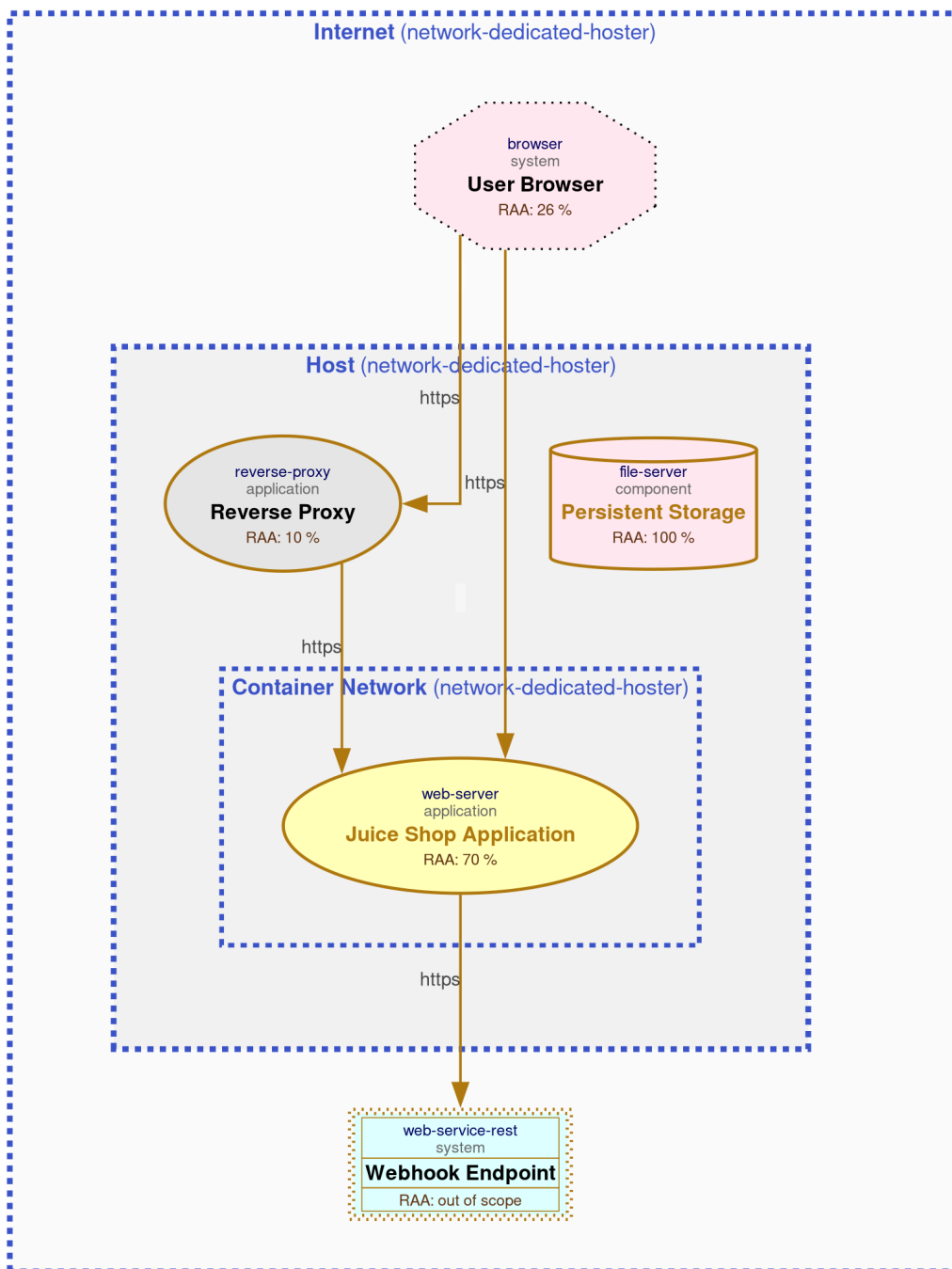
Security Requirements

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

AuthZ on sensitive routes
Implement strict server-side authorization checks (role/permission) on admin or sensitive functionalities.

Rate limiting & lockouts
Apply rate limiting and account lockout policies to mitigate brute-force and automated attacks on authentication and expensive operations.

Secrets management
Protect secret keys and credentials (JWT signing keys, OAuth client secrets) – keep them out of code repos and avoid logging them.

Secure headers
Add security headers (HSTS, CSP, X-Frame-Options, X-Content-Type-Options, etc.) at the proxy or app to mitigate client-side attacks.

TLS in transit
Enforce HTTPS for user traffic via a TLS-terminating reverse proxy with strong ciphers and certificate management.

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.

Credential Stuffing / Brute Force
Attackers attempt repeated login attempts to guess credentials or exhaust system resources.

SSRF via Outbound Requests
Server-side requests (e.g. profile image URL fetch or WebHook callback) are abused to access internal network resources.

Stored XSS via Product Reviews
Malicious scripts are inserted into product reviews, getting stored and executed in other users'™ browsers.

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.

actor
User Browser

app
Juice Shop Application

auth
Tokens & Sessions, User Accounts

direct
Direct to App (no proxy)

docker
Docker Host

egress
To Challenge WebHook

logs
Logs

nodejs
Juice Shop Application

optional
Reverse Proxy

pii
Orders, User Accounts

primary
To Reverse Proxy (preferred)

proxy
Reverse Proxy

public
Product Catalog

saas
Webhook Endpoint

storage
Persistent Storage

tokens
Tokens & Sessions

user
User Browser

volume
Persistent Storage

webhook
Webhook Endpoint

STRIDE Classification of Identified Risks

This chapter clusters and classifies the risks by STRIDE categories: In total **20 potential risks** have been identified during the threat modeling process of which **3 in the Spoofing** category, **6 in the Tampering** category, **0 in the Repudiation** category, **4 in the Information Disclosure** category, **0 in the Denial of Service** category, and **7 in the Elevation of Privilege** category.

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

Spoofing

Medium: **Cross-Site Request Forgery (CSRF)**: 2 / 2 Risks - Exploitation likelihood is *Very Likely* with *Low* impact.
When a web application is accessed via web protocols Cross-Site Request Forgery (CSRF) risks might arise.

Medium: **Missing Identity Store**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
The modeled architecture does not contain an identity store, which might be the risk of a model missing critical assets (and thus not seeing their risks).

Tampering

Elevated: **Cross-Site Scripting (XSS)**: 1 / 1 Risk - Exploitation likelihood is *Likely* with *Medium* impact.
For each web application Cross-Site Scripting (XSS) risks might arise. In terms of the overall risk level take other applications running on the same domain into account as well.

Medium: **Container Base Image Backdooring**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* 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 Build Infrastructure**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.
The modeled architecture does not contain a build infrastructure (devops-client, sourcecode-repo, build-pipeline, etc.), which might be the risk of a model missing critical assets (and thus not seeing their risks). If the architecture contains custom-developed parts, the pipeline where code gets developed and built needs to be part of the model.

Medium: **Missing Hardening**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* impact.
Technical assets with a Relative Attacker Attractiveness (RAA) value of 55 % or higher should be explicitly hardened taking best practices and vendor hardening guides into account.

Low: **Missing Web Application Firewall (WAF)**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Low* 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.

Repudiation

n/a

Information Disclosure

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

Medium: **Server-Side Request Forgery (SSRF)**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* 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.

Medium: **Unencrypted Technical Assets**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* 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

n/a

Elevation of Privilege

Elevated: **Missing Authentication**: 1 / 1 Risk - 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.

Medium: **Missing Two-Factor Authentication (2FA)**: 2 / 2 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.

Low: **Unnecessary Data Transfer**: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
When a technical asset sends or receives data assets, which it neither processes or stores this is an indicator for unnecessarily transferred data (or for an incomplete model). When the unnecessarily transferred data assets are sensitive, this poses an unnecessary risk of an increased attack surface.

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

Assignment by Function

This chapter clusters and assigns the risks by functions which are most likely able to check and mitigate them: In total **20 potential risks** have been identified during the threat modeling process of which **2 should be checked by Business Side**, **8 should be checked by Architecture**, **5 should be checked by Development**, and **5 should be checked by Operations**.

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

Business Side

Medium: **Missing Two-Factor Authentication (2FA)**: 2 / 2 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**: 1 / 1 Risk - 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.

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

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

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

Low: **Unnecessary Data Transfer**: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
Try to avoid sending or receiving sensitive data assets which are not required (i.e. neither processed or stored) by the involved technical asset.

Low: **Unnecessary Technical Asset**: 2 / 2 Risks - Exploitation likelihood is *Unlikely* with *Low* impact.
Try to avoid using technical assets that do not process or store anything.

Development

Elevated: **Cross-Site Scripting (XSS)**: 1 / 1 Risk - Exploitation likelihood is *Likely* with *Medium* impact.

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

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

Try to use anti-CSRF tokens ot the double-submit patterns (at least for logged-in requests). When your authentication scheme depends on cookies (like session or token cookies), consider marking them with the same-site flag. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

Medium: **Server-Side Request Forgery (SSRF)**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* 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.

Operations

Medium: **Container Base Image Backdooring**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* 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 Hardening**: 2 / 2 Risks - Exploitation likelihood is *Likely* with *Low* impact.

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

Medium: **Unencrypted Technical Assets**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Medium* impact.

Apply encryption to the technical asset.

Low: **Missing Web Application Firewall (WAF)**: 1 / 1 Risk - Exploitation likelihood is *Unlikely* with *Low* 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 enhances by a WAF component via ModSecurity plugins.

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.

Persistent Storage: RAA 100%
Host-mounted volume for database, file uploads, and logs.

Juice Shop Application: RAA 70%
OWASP Juice Shop server (Node.js/Express, v19.0.0).

User Browser: RAA 26%
End-user web browser (client).

Reverse Proxy: RAA 10%
Optional reverse proxy (e.g., Nginx) for TLS termination and adding security headers.

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.
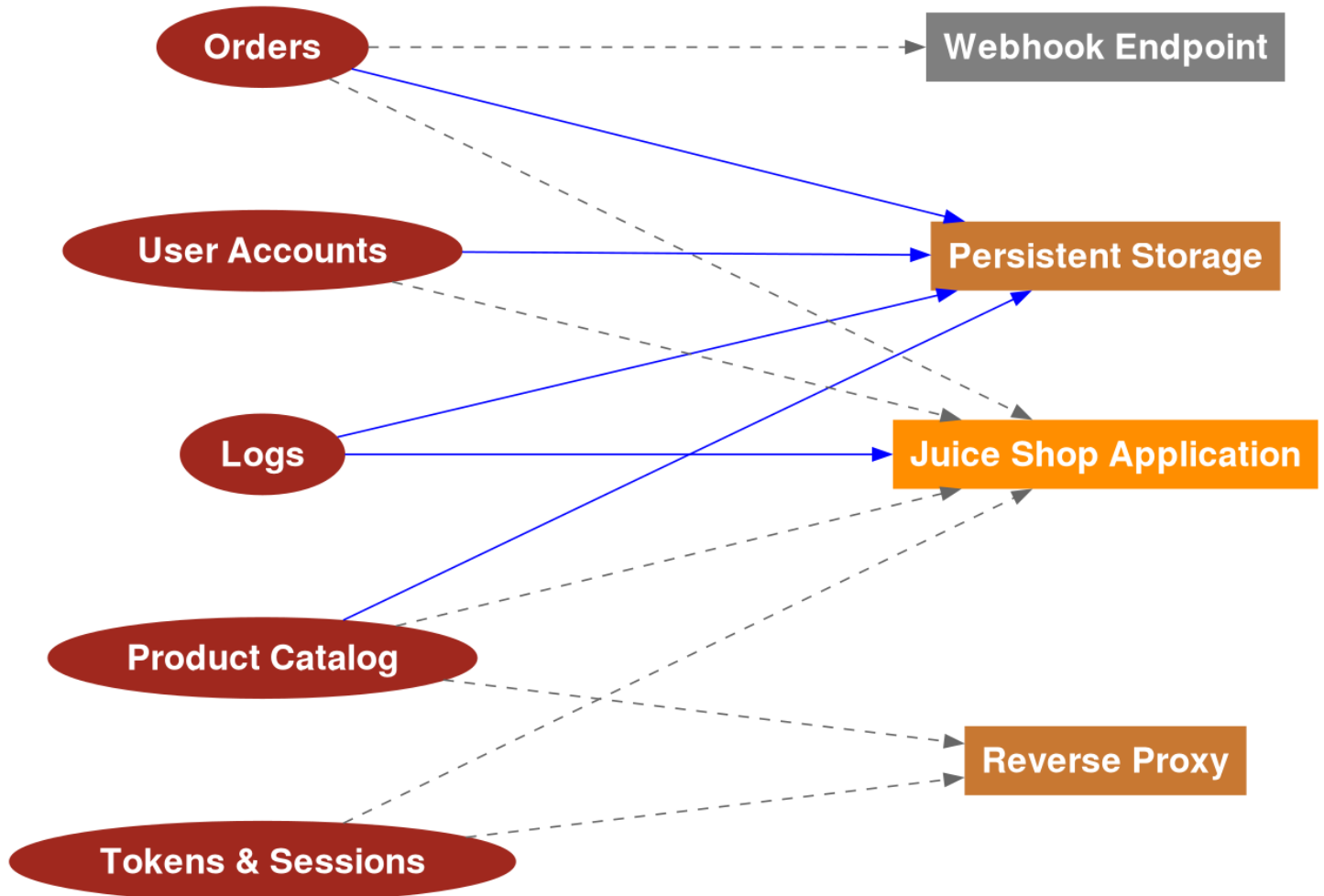
Out-of-Scope Assets: 1 Asset

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.

Webhook Endpoint: out-of-scope
Third-party service to receive notifications (not under our control).

Potential Model Failures: 7 / 7 Risks

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

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

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

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

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

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

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

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

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

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

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

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).

Questions: 4 / 4 Questions

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

Are any outbound integrations (webhooks) configured?
- answer pending -

Do you expose port 3000 beyond localhost?
- answer pending -

Do you use a reverse proxy with TLS and security headers?
- answer pending -

Is any sensitive data stored in logs or files?
- answer pending -

Identified Risks by Vulnerability Category

In total **20 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **2 as elevated**, **13 as medium**, and **5 as low**.

These risks are distributed across **14 vulnerability categories**. The following sub-chapters of this section describe each identified risk category.

Cross-Site Scripting (XSS): 1 / 1 Risk

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 **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

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

cross-site-scripting@juice-shop

Unchecked

Missing Authentication: 1 / 1 Risk

Description (Elevation of Privilege): <u>CWE 306</u>

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: <u>V2 - Authentication Verification Requirements</u>
Cheat Sheet: <u>Authentication_Cheat_Sheet</u>

Check

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

Risk Findings

The risk **Missing Authentication** 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 Authentication covering communication link **To App** from **Reverse Proxy** to **Juice Shop Application**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Unchecked

Container Base Image Backdooring: 1 / 1 Risk

Description (Tampering): CWE 912

When a technical asset is built using container technologies, Base Image Backdooring risks might arise where base images and other layers used contain vulnerable components or backdoors.

See for example:
https://techcrunch.com/2018/06/15/tainted-crypto-mining-containers-pulled-from-docker-hub/

Impact

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

Detection Logic

In-scope technical assets running as containers.

Risk Rating

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

False Positives

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

Mitigation (Operations): Container Infrastructure Hardening

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

ASVS Chapter: V10 - Malicious Code Verification Requirements
Cheat Sheet: Docker_Security_Cheat_Sheet

Check

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

Risk Findings

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

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

Medium Risk Severity

Container Base Image Backdooring risk at **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

container-baseimage-backdooring@juice-shop

Unchecked

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

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

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

Impact

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

Detection Logic

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

Risk Rating

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

False Positives

Web applications passing the authentication sate 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 ot the double-submit patterns (at least for logged-in requests). When your authentication scheme depends on cookies (like session or token cookies), consider marking them with the same-site flag. When a third-party product is used instead of custom developed software, check if the product applies the proper mitigation and ensure a reasonable patch-level.

ASVS Chapter: [V4 - Access Control Verification Requirements](#)
Cheat Sheet: [Cross-Site_Request_Forgery_Prevention_Cheat_Sheet](#)

Check

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

Risk Findings

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

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

Medium Risk Severity

Cross-Site Request Forgery (CSRF) risk at **Juice Shop Application** via **Direct to App (no proxy)** from **User Browser**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Unchecked

Cross-Site Request Forgery (CSRF) risk at **Juice Shop Application** via **To App** from **Reverse Proxy**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Unchecked

Missing Build Infrastructure: 1 / 1 Risk

Description (Tampering): CWE 1127

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

Impact

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

Detection Logic

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

Risk Rating

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

False Positives

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

Mitigation (Architecture): Build Pipeline Hardening

Include the build infrastructure in the model.

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

Check

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

Risk Findings

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

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

Medium Risk Severity

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

missing-build-infrastructure@juice-shop

Unchecked

Missing Hardening: 2 / 2 Risks

Description (Tampering): CWE 16

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

Impact

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

Detection Logic

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

Risk Rating

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

False Positives

Usually no false positives.

Mitigation (Operations): System Hardening

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

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

Check

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

Risk Findings

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

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

Medium Risk Severity

Missing Hardening risk at **Juice Shop Application**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@juice-shop

Unchecked

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

missing-hardening@persistent-storage

Unchecked

Missing Identity Store: 1 / 1 Risk

Description (Spoofing): CWE 287

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

Impact

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

Detection Logic

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

Risk Rating

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

False Positives

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

Mitigation (Architecture): Identity Store

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

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

Check

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

Risk Findings

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

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

Medium Risk Severity

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

missing-identity-store@reverse-proxy

Unchecked

Missing Two-Factor Authentication (2FA): 2 / 2 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 **2 times** in the analyzed architecture to be potentially possible. Each spot should be checked individually by reviewing the implementation whether all controls have been applied properly in order to mitigate each risk.

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

Medium Risk Severity

Missing Two-Factor Authentication covering communication link **Direct to App (no proxy)** from **User Browser** to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Unchecked

Missing Two-Factor Authentication covering communication link **To App** from **User Browser** forwarded via **Reverse Proxy** to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

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 **Juice Shop Application** as an example): Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-vault@juice-shop

Unchecked

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

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

Medium Risk Severity

Server-Side Request Forgery (SSRF) risk at **Juice Shop Application** server-side web-requesting the target **Webhook Endpoint** via **To Challenge WebHook**: Exploitation likelihood is *Likely* with *Low* impact.

server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Unchecked

Server-Side Request Forgery (SSRF) risk at **Reverse Proxy** server-side web-requesting the target **Juice Shop Application** via **To App**: Exploitation likelihood is *Likely* with *Low* impact.

server-side-request-forgery@reverse-proxy@juice-shop@reverse-proxy>to-app

Unchecked

Unencrypted Technical Assets: 1 / 1 Risk

Description (Information Disclosure): <u>CWE 311</u>

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: <u>V6 - Stored Cryptography Verification Requirements</u>
Cheat Sheet: <u>Cryptographic_Storage_Cheat_Sheet</u>

Check

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

Risk Findings

The risk **Unencrypted Technical Assets** 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

Unencrypted Technical Asset named **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@juice-shop

Unchecked

Missing Web Application Firewall (WAF): 1 / 1 Risk

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

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

Low Risk Severity

Missing Web Application Firewall (WAF) risk at **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Low* impact.

missing-waf@juice-shop

Unchecked

Unnecessary Data Transfer: 2 / 2 Risks

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

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

Impact

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

Detection Logic

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

Risk Rating

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

False Positives

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

Mitigation (Architecture): Attack Surface Reduction

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

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

Check

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

Risk Findings

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

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

Low Risk Severity

Unnecessary Data Transfer of **Tokens & Sessions** data at **User Browser** from/to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@tokens-sessions@user-browser@juice-shop

Unchecked

Unnecessary Data Transfer of **Tokens & Sessions** data at **User Browser** from/to **Reverse Proxy**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@tokens-sessions@user-browser@reverse-proxy

Unchecked

Unnecessary Technical Asset: 2 / 2 Risks

Description (Elevation of Privilege): CWE 1008

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).

Impact

If this risk is unmitigated, attackers might be able to target unnecessary technical assets.

Detection Logic

Technical assets not processing or storing any data assets.

Risk Rating

low

False Positives

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

Mitigation (Architecture): Attack Surface Reduction

Try to avoid using technical assets that do not process or store anything.

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

Check

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

Risk Findings

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

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

Low Risk Severity

Unnecessary Technical Asset named **Persistent Storage**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-technical-asset@persistent-storage

Unchecked

Unnecessary Technical Asset named **User Browser**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-technical-asset@user-browser

Unchecked

Identified Risks by Technical Asset

In total **20 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **2 as elevated**, **13 as medium**, and **5 as low**.

These risks are distributed across **4 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.

Juice Shop Application: 13 / 13 Risks

Description

OWASP Juice Shop server (Node.js/Express, v19.0.0).

Identified Risks of Asset

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

Elevated Risk Severity

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

cross-site-scripting@juice-shop

> **Unchecked**

Missing Authentication covering communication link **To App** from **Reverse Proxy** to **Juice Shop Application**: Exploitation likelihood is *Likely* with *Medium* impact.

missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

> **Unchecked**

Medium Risk Severity

Container Base Image Backdooring risk at **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

container-baseimage-backdooring@juice-shop

> **Unchecked**

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

missing-build-infrastructure@juice-shop

> **Unchecked**

Missing Two-Factor Authentication covering communication link **Direct to App (no proxy)** from **User Browser** to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

> **Unchecked**

Missing Two-Factor Authentication covering communication link **To App** from **User Browser** forwarded via **Reverse Proxy** to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

> **Unchecked**

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

missing-vault@juice-shop

Unchecked

Unencrypted Technical Asset named **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Medium* impact.

unencrypted-asset@juice-shop

Unchecked

Cross-Site Request Forgery (CSRF) risk at **Juice Shop Application** via **Direct to App (no proxy)** from **User Browser**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Unchecked

Cross-Site Request Forgery (CSRF) risk at **Juice Shop Application** via **To App** from **Reverse Proxy**: Exploitation likelihood is *Very Likely* with *Low* impact.

cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Unchecked

Missing Hardening risk at **Juice Shop Application**: Exploitation likelihood is *Likely* with *Low* impact.

missing-hardening@juice-shop

Unchecked

Server-Side Request Forgery (SSRF) risk at **Juice Shop Application** server-side web-requesting the target **Webhook Endpoint** via **To Challenge WebHook**: Exploitation likelihood is *Likely* with *Low* impact.

server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Unchecked

Low Risk Severity

Missing Web Application Firewall (WAF) risk at **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Low* impact.

missing-waf@juice-shop

Unchecked

Asset Information

ID:	juice-shop
Type:	process
Usage:	business
RAA:	70 %
Size:	application
Technology:	web-server

Tags:	app, nodejs
Internet:	false
Machine:	container
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	true
Client by Human:	false
Data Processed:	Orders, Product Catalog, Tokens & Sessions, User Accounts
Data Stored:	Logs
Formats Accepted:	JSON

Asset Rating

Owner:	Lab Owner	
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:	In-scope web application (contains all business logic and vulnerabilities by design).	

Outgoing Communication Links: 1

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

To Challenge WebHook (outgoing)

Optional outbound callback (HTTP POST) to external WebHook when a challenge is solved.

Target:	Webhook Endpoint
Protocol:	https
Encrypted:	true
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	egress
VPN:	false
IP-Filtered:	false
Data Sent:	Orders

Data Received: none

Incoming Communication Links: 2
Source technical asset names are clickable and link to the corresponding chapter.

Direct to App (no proxy) (incoming)
Direct browser access to app (HTTP on 3000).

Source:	User Browser
Protocol:	https
Encrypted:	true
Authentication:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false
Usage:	business
Tags:	direct
VPN:	false
IP-Filtered:	false
Data Received:	Tokens & Sessions
Data Sent:	Product Catalog

To App (incoming)
Proxy forwarding to app (HTTP on 3000 internally).

Source:	Reverse Proxy
Protocol:	https
Encrypted:	true
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Received:	Tokens & Sessions
Data Sent:	Product Catalog

Persistent Storage: 2 / 2 Risks

Description

Host-mounted volume for database, file uploads, and logs.

Identified Risks of Asset

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

Medium Risk Severity

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

missing-hardening@persistent-storage

Unchecked

Low Risk Severity

Unnecessary Technical Asset named **Persistent Storage**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-technical-asset@persistent-storage

Unchecked

Asset Information

ID:	persistent-storage
Type:	datastore
Usage:	devops
RAA:	100 %
Size:	component
Technology:	file-server
Tags:	storage, volume
Internet:	false
Machine:	virtual
Encryption:	transparent
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	none
Data Stored:	Logs, Orders, Product Catalog, User Accounts

Formats Accepted: File

Asset Rating

Owner: Lab Owner
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: Local disk storage for the container – not directly exposed, but if
                        compromised it contains sensitive data (database and logs).

Reverse Proxy: 2 / 2 Risks

Description

Optional reverse proxy (e.g., Nginx) for TLS termination and adding security headers.

Identified Risks of Asset

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

Medium Risk Severity

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

missing-identity-store@reverse-proxy

Unchecked

Server-Side Request Forgery (SSRF) risk at **Reverse Proxy** server-side web-requesting the target **Juice Shop Application** via **To App**: Exploitation likelihood is *Likely* with *Low* impact.

server-side-request-forgery@reverse-proxy@juice-shop@reverse-proxy>to-app

Unchecked

Asset Information

ID:	reverse-proxy
Type:	process
Usage:	business
RAA:	10 %
Size:	application
Technology:	reverse-proxy
Tags:	optional, proxy
Internet:	false
Machine:	virtual
Encryption:	transparent
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	false
Data Processed:	Product Catalog, Tokens & Sessions
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	Lab Owner	
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:	Not exposed to internet directly; improves security of inbound traffic.	

Outgoing Communication Links: 1
Target technical asset names are clickable and link to the corresponding chapter.

To App (outgoing)
Proxy forwarding to app (HTTP on 3000 internally).

Target:	Juice Shop Application
Protocol:	https
Encrypted:	true
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	none
VPN:	false
IP-Filtered:	false
Data Sent:	Tokens & Sessions
Data Received:	Product Catalog

Incoming Communication Links: 1
Source technical asset names are clickable and link to the corresponding chapter.

To Reverse Proxy (preferred) (incoming)
User browser to reverse proxy (HTTPS on 443).

Source:	User Browser
Protocol:	https
Encrypted:	true
Authentication:	session-id
Authorization:	enduser-identity-propagation
Read-Only:	false

Usage:	business
Tags:	primary
VPN:	false
IP-Filtered:	false
Data Received:	Tokens & Sessions
Data Sent:	Product Catalog

User Browser: 3 / 3 Risks

Description

End-user web browser (client).

Identified Risks of Asset

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

Low Risk Severity

Unnecessary Data Transfer of **Tokens & Sessions** data at **User Browser** from/to **Juice Shop Application**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@tokens-sessions@user-browser@juice-shop

> **Unchecked**

Unnecessary Data Transfer of **Tokens & Sessions** data at **User Browser** from/to **Reverse Proxy**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-data-transfer@tokens-sessions@user-browser@reverse-proxy

> **Unchecked**

Unnecessary Technical Asset named **User Browser**: Exploitation likelihood is *Unlikely* with *Low* impact.

unnecessary-technical-asset@user-browser

> **Unchecked**

Asset Information

ID:	user-browser
Type:	external-entity
Usage:	business
RAA:	26 %
Size:	system
Technology:	browser
Tags:	actor, user
Internet:	true
Machine:	virtual
Encryption:	none
Multi-Tenant:	false
Redundant:	false
Custom-Developed:	false
Client by Human:	true

Data Processed: none
Data Stored: none
Formats Accepted: JSON

Asset Rating

Owner: External User
Confidentiality: public   (rated 1 in scale of 5)
Integrity: operational   (rated 2 in scale of 5)
Availability: operational   (rated 2 in scale of 5)
CIA-Justification: Client controlled by end user (potentially an attacker).

Outgoing Communication Links: 2

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

To Reverse Proxy (preferred) (outgoing)

User browser to reverse proxy (HTTPS on 443).

Target: Reverse Proxy
Protocol: https
Encrypted: true
Authentication: session-id
Authorization: enduser-identity-propagation
Read-Only: false
Usage: business
Tags: primary
VPN: false
IP-Filtered: false
Data Sent: Tokens & Sessions
Data Received: Product Catalog

Direct to App (no proxy) (outgoing)

Direct browser access to app (HTTP on 3000).

Target: Juice Shop Application
Protocol: https
Encrypted: true
Authentication: session-id
Authorization: enduser-identity-propagation

Read-Only: false
Usage: business
Tags: direct
VPN: false
IP-Filtered: false
Data Sent: Tokens & Sessions
Data Received: Product Catalog

Webhook Endpoint: out-of-scope

Description

External WebHook service (3rd-party, if configured for integrations).

Identified Risks of Asset

Asset was defined as out-of-scope.

Asset Information

ID:	webhook-endpoint
Type:	external-entity
Usage:	business
RAA:	out-of-scope
Size:	system
Technology:	web-service-rest
Tags:	saas, webhook
Internet:	true
Machine:	virtual
Encryption:	none
Multi-Tenant:	true
Redundant:	true
Custom-Developed:	false
Client by Human:	false
Data Processed:	Orders
Data Stored:	none
Formats Accepted:	JSON

Asset Rating

Owner:	Third-Party	
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:	External service that receives data (like order or challenge info). Treated as a trusted integration point but could be abused if misconfigured.	

Asset Out-of-Scope Justification

Third-party service to receive notifications (not under our control).

Incoming Communication Links: 1
Source technical asset names are clickable and link to the corresponding chapter.

To Challenge WebHook (incoming)

Optional outbound callback (HTTP POST) to external WebHook when a challenge is solved.

Source:	Juice Shop Application
Protocol:	https
Encrypted:	true
Authentication:	none
Authorization:	none
Read-Only:	false
Usage:	business
Tags:	egress
VPN:	false
IP-Filtered:	false
Data Received:	Orders
Data Sent:	none

Identified Data Breach Probabilities by Data Asset

In total **20 potential risks** have been identified during the threat modeling process of which **0 are rated as critical**, **0 as high**, **2 as elevated**, **13 as medium**, and **5 as low**.

These risks are distributed across **5 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.

Logs: 13 / 13 Risks

Application and access logs (may inadvertently contain PII or secrets).

ID:	logs
Usage:	devops
Quantity:	many
Tags:	logs
Origin:	application
Owner:	Lab Owner
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:	Logs are for internal use (troubleshooting, monitoring). They should not be exposed publicly, and sensitive data should be sanitized to protect confidentiality.
Processed by:	none
Stored by:	Juice Shop Application, Persistent Storage
Sent via:	none
Received via:	none
Data Breach:	**probable**
Data Breach Risks:	This data asset has data breach potential because of 13 remaining risks:

Probable: container-baseimage-backdooring@juice-shop

Possible: cross-site-scripting@juice-shop

Possible: missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Possible: missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Improbable: cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Improbable: cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Improbable: missing-hardening@juice-shop

Improbable: missing-hardening@persistent-storage

Improbable: missing-waf@juice-shop

Improbable: unencrypted-asset@juice-shop

Improbable: unnecessary-technical-asset@persistent-storage

Orders: 13 / 13 Risks

Order history, addresses, and payment metadata (no raw card numbers).

ID:	orders
Usage:	business
Quantity:	many
Tags:	pii
Origin:	application
Owner:	Lab Owner
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:	Contains users' personal data and business transaction records. Integrity and confidentiality are important to prevent fraud or privacy breaches.
Processed by:	Juice Shop Application, Webhook Endpoint
Stored by:	Persistent Storage
Sent via:	To Challenge WebHook
Received via:	none
Data Breach:	**probable**
Data Breach Risks:	This data asset has data breach potential because of 13 remaining risks:

Probable: container-baseimage-backdooring@juice-shop

Possible: cross-site-scripting@juice-shop

Possible: missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Possible: missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Improbable: cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Improbable: cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Improbable: missing-hardening@juice-shop

Improbable: missing-hardening@persistent-storage

Improbable: missing-waf@juice-shop

Improbable: unencrypted-asset@juice-shop

Improbable: unnecessary-technical-asset@persistent-storage

Product Catalog: 14 / 14 Risks

Product information (names, descriptions, prices) available to all users.

ID:	product-catalog
Usage:	business
Quantity:	many
Tags:	public
Origin:	application
Owner:	Lab Owner
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: Product data is intended to be public, but its integrity is important (to avoid defacement or price manipulation that could mislead users).

Processed by: Juice Shop Application, Reverse Proxy
Stored by: Persistent Storage
Sent via: none
Received via: To Reverse Proxy (preferred), To App, Direct to App (no proxy)
Data Breach: **probable**
Data Breach Risks: This data asset has data breach potential because of 14 remaining risks:

Probable: container-baseimage-backdooring@juice-shop

Possible: cross-site-scripting@juice-shop

Possible: missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Possible: missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Possible: server-side-request-forgery@reverse-proxy@juice-shop@reverse-proxy>to-app

Improbable: cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Improbable: cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Improbable: missing-hardening@juice-shop

Improbable: missing-hardening@persistent-storage

Improbable: missing-waf@juice-shop

Improbable: unencrypted-asset@juice-shop

Improbable: unnecessary-technical-asset@persistent-storage

Tokens & Sessions: 12 / 12 Risks

Session identifiers, JWTs for authenticated sessions, CSRF tokens.

ID:	tokens-sessions
Usage:	business
Quantity:	many
Tags:	auth, tokens
Origin:	application
Owner:	Lab Owner
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:	If session tokens are compromised, attackers can hijack user sessions. They must be kept confidential and intact; availability is less critical (tokens can be reissued).
Processed by:	Juice Shop Application, Reverse Proxy
Stored by:	none
Sent via:	To Reverse Proxy (preferred), To App, Direct to App (no proxy)
Received via:	none
Data Breach:	**probable**
Data Breach Risks:	This data asset has data breach potential because of 12 remaining risks:

Probable: container-baseimage-backdooring@juice-shop

Possible: cross-site-scripting@juice-shop

Possible: missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Possible: missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Possible: server-side-request-forgery@reverse-proxy@juice-shop@reverse-proxy>to-app

Improbable: cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Improbable: cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Improbable: missing-hardening@juice-shop

Improbable: missing-waf@juice-shop

Improbable: unencrypted-asset@juice-shop

User Accounts: 13 / 13 Risks

User profile data, credential hashes, emails.

ID:	user-accounts	
Usage:	business	
Quantity:	many	
Tags:	auth, pii	
Origin:	user-supplied	
Owner:	Lab Owner	
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:	Contains personal identifiers and authentication data. High confidentiality is required to protect user privacy, and integrity is critical to prevent account takeovers.	
Processed by:	Juice Shop Application	
Stored by:	Persistent Storage	
Sent via:	none	
Received via:	none	
Data Breach:	**probable**	

Data Breach Risks: This data asset has data breach potential because of 13 remaining risks:

Probable: container-baseimage-backdooring@juice-shop

Possible: cross-site-scripting@juice-shop

Possible: missing-authentication@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: missing-authentication-second-factor@user-browser>direct-to-app-no-proxy@user-browser@juice-shop

Possible: missing-authentication-second-factor@reverse-proxy>to-app@reverse-proxy@juice-shop

Possible: server-side-request-forgery@juice-shop@webhook-endpoint@juice-shop>to-challenge-webhook

Improbable: cross-site-request-forgery@juice-shop@user-browser>direct-to-app-no-proxy

Improbable: cross-site-request-forgery@juice-shop@reverse-proxy>to-app

Improbable: missing-hardening@juice-shop

Improbable: missing-hardening@persistent-storage

Improbable: missing-waf@juice-shop

Improbable: unencrypted-asset@juice-shop

Improbable: unnecessary-technical-asset@persistent-storage

Trust Boundaries

In total **3 trust boundaries** have been modeled during the threat modeling process.

Container Network
Docker container network (isolated internal network for containers).

ID:	container-network
Type:	network-dedicated-hoster
Tags:	none
Assets inside:	Juice Shop Application
Boundaries nested:	none

Host
Local host machine / VM running the Docker environment.

ID:	host
Type:	network-dedicated-hoster
Tags:	none
Assets inside:	Persistent Storage, Reverse Proxy
Boundaries nested:	Container Network

Internet
Untrusted public network (Internet).

ID:	internet
Type:	network-dedicated-hoster
Tags:	none
Assets inside:	User Browser, Webhook Endpoint
Boundaries nested:	Host

Shared Runtimes

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

Docker Host
Docker Engine and default bridge network on the host.

ID:	docker-host
Tags:	docker
Assets running:	Juice Shop Application

Risk Rules Checked by Threagile

Threagile Version: 1.0.0

Threagile Build Timestamp: 20240730113903

Threagile Execution Timestamp: 20260212143423

Model Filename: /app/work/labs/lab2/threagile-model.secure.yaml

Model Hash (SHA256): 2faa67841a25406d4ce6e5d023b635aa78428cbcc0c8f2562d6625b7eea69534

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

Accidental Secret Leak
accidental-secret-leak

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

Code Backdooring
code-backdooring

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

Container Base Image Backdooring
container-baseimage-backdooring

STRIDE: Tampering

Description: When a technical asset is built using container technologies, Base Image
                Backdooring risks might arise where base images and other layers used contain
                vulnerable components or backdoors.

Detection: In-scope technical assets running as containers.

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

Container Platform Escape
container-platform-escape

STRIDE: Elevation of Privilege

Description: Container platforms are especially interesting targets for attackers as they host big
                parts of a containerized runtime infrastructure. When not configured and operated
                with security best practices in mind, attackers might exploit a vulnerability inside an
                container and escape towards the platform as highly privileged users. These
                scenarios might give attackers capabilities to attack every other container as owning
                the container platform (via container escape attacks) equals to owning every
                container.

Detection: In-scope container platforms.

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

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

STRIDE: Spoofing

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

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

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

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

STRIDE: Tampering

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

Detection: In-scope web applications.

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

DoS-risky Access Across Trust-Boundary

dos-risky-access-across-trust-boundary

STRIDE: Denial of Service

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

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

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

Incomplete Model
incomplete-model

STRIDE: Information Disclosure

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

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

Rating: low

LDAP-Injection
ldap-injection

STRIDE: Tampering

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

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

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

Missing Authentication
missing-authentication

STRIDE: Elevation of Privilege

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

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

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

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

Missing Two-Factor Authentication (2FA)
missing-authentication-second-factor

STRIDE: Elevation of Privilege

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

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

Rating: medium

Missing Build Infrastructure
missing-build-infrastructure

STRIDE: Tampering

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

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

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

Missing Cloud Hardening
missing-cloud-hardening

STRIDE: Tampering

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

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

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

Missing File Validation
missing-file-validation

STRIDE: Spoofing

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

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

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

Missing Hardening
missing-hardening

STRIDE: Tampering

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

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

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

Missing Identity Propagation
missing-identity-propagation

STRIDE: Elevation of Privilege

Description: Technical assets (especially multi-tenant systems), which usually process data for endusers should authorize every request based on the identity of the enduser when the data flow is authenticated (i.e. non-public). For DevOps usages at least a technical-user authorization is required.

Detection: In-scope service-like technical assets which usually process data based on enduser requests, if authenticated (i.e. non-public), should authorize incoming requests based on the propagated enduser identity when their rating is sensitive. This is especially the case for all multi-tenant assets (there even less-sensitive rated ones). DevOps usages are exempted from this risk.

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

Missing Identity Provider Isolation
missing-identity-provider-isolation

STRIDE: Elevation of Privilege

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

Detection: In-scope identity provider assets and their identity datastores when surrounded by other (not identity-related) assets (without a network trust-boundary in-between).

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

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

Missing Identity Store
missing-identity-store

STRIDE: Spoofing

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

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

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

Missing Network Segmentation
missing-network-segmentation

STRIDE: Elevation of Privilege

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

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

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

Missing Vault (Secret Storage)
missing-vault

STRIDE: Information Disclosure

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

Detection: Models without a Vault (Secret Storage).

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

Missing Vault Isolation
missing-vault-isolation

STRIDE: Elevation of Privilege

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

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

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

Missing Web Application Firewall (WAF)
missing-waf

STRIDE: Tampering

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

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

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

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

STRIDE: Elevation of Privilege

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

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

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

Path-Traversal
path-traversal

STRIDE: Information Disclosure

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

and of the data assets processed or stored.

Detection: Filesystems accessed by in-scope callers.

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

Push instead of Pull Deployment
push-instead-of-pull-deployment

STRIDE: Tampering

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

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

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

Search-Query Injection
search-query-injection

STRIDE: Tampering

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

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

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

Server-Side Request Forgery (SSRF)
server-side-request-forgery

STRIDE: Information Disclosure

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

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

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

Service Registry Poisoning

service-registry-poisoning

STRIDE: Spoofing

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

Detection: In-scope service registries.

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

SQL/NoSQL-Injection
sql-nosql-injection

STRIDE: Tampering

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

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

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

Unchecked Deployment
unchecked-deployment

STRIDE: Tampering

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

Detection: All development-relevant technical assets.

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

Unencrypted Technical Assets
unencrypted-asset

STRIDE: Information Disclosure

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

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

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

Unencrypted Communication
unencrypted-communication

STRIDE: Information Disclosure

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

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

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

Unguarded Access From Internet
unguarded-access-from-internet

STRIDE: Elevation of Privilege

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

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

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

Unguarded Direct Datastore Access
unguarded-direct-datastore-access

STRIDE: Elevation of Privilege

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

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

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

Unnecessary Communication Link
unnecessary-communication-link

STRIDE: Elevation of Privilege

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

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

Rating: low

Unnecessary Data Asset
unnecessary-data-asset

STRIDE: Elevation of Privilege

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

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

Rating: low

Unnecessary Data Transfer
unnecessary-data-transfer

STRIDE: Elevation of Privilege

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

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

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

Unnecessary Technical Asset
unnecessary-technical-asset

STRIDE: Elevation of Privilege

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

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

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

Rating: low

Untrusted Deserialization
untrusted-deserialization

STRIDE: Tampering

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

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

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

Wrong Communication Link Content
wrong-communication-link-content

STRIDE: Information Disclosure

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

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

Rating: low

Wrong Trust Boundary Content
wrong-trust-boundary-content

STRIDE: Elevation of Privilege

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

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

Rating: low

XML External Entity (XXE)
xml-external-entity

STRIDE: Information Disclosure

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

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

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

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

Disclaimer

Student Name 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. Student Name 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 Student Name 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.