Threat Modeling Report Using Microsoft Threat Modeling

(SQL Database & Web Application Component)

Prepared and Reported by: Bamidele Olaleke

Date: November 2025

1. Executive Summary

A threat-modeling assessment was conducted on the system architecture involving the **SQL Database**, **Web Server**, and associated data flows (HTTP requests, responses, SQL operations). The assessment identified **17 threats**, all marked as **Mitigation Implemented** in the source model.
The threats span all STRIDE categories—Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

This report consolidates these threats, assesses risks, and provides professional recommendations to ensure secure design and continuous improvement.

2. Threat Model Overview

Summary of Findings

Status	Count
Mitigation Implemented	**17**
Not Started	0
Not Applicable	0
Needs Investigation	0
Total Threats	**17**

The model suggests strong progress toward mitigation; however, some threats still show **“no mitigation provided”** in their justification field, indicating incomplete implementation details.

3. Key Threats Identified

Below is a summarized representation of the key threats extracted directly from the OWASP Threat Dragon report

3.1 Spoofing Threats

Spoofing of Source Data Store – SQL Database

- **Category:** Spoofing
- **Description:** SQL Database could be impersonated, leading to incorrect data delivered to the web server.
- **Risk:** High (could redirect or poison data flows)
- **Mitigation:** Enforce **mutual TLS** authentication between web server ↔ SQL database

Spoofing of Destination Data Store – SQL Database

- **Description:** Data intended for SQL Database may be redirected to an attacker-controlled datastore.
- **Mitigation:** Enforce **firewall rules** restricting outbound DB traffic

---

3.2 Tampering Threats

Persistent Cross-Site Scripting (Stored XSS)

- **Category:** Tampering
- **Description:** SQL Database content is not sanitized, allowing persistent malicious scripts.
- **Mitigation:** Sanitize database input via libraries such as DOMPurify

Cross-Site Scripting (Reflected XSS)

- **Category:** Tampering
- **Description:** Untrusted input is not sanitized on the web server.
- **Mitigation:** Implemented.

Risks from Logging

- **Category:** Tampering
- **Description:** Log files could be weaponized against log readers.

- **Mitigation:** Implemented

3.3 Repudiation Threats

Lower-Trusted Subject Updates Logs

- **Category:** Repudiation
- **Description:** Lower-privileged actors can modify logs, creating accountability issues.
- **Mitigation:** Documented.

Insufficient Auditing

- **Category:** Repudiation
- **Description:** Logs may not capture enough detail to investigate incidents.
- **Mitigation:** None documented.

3.4 Information Disclosure Threats

8. Authorization Bypass – SQL Database

- **Category:** Information Disclosure
- **Description:** Attackers may bypass permissions via direct file access or file-sharing.
- **Mitigation:** None documented.

Weak Access Control for SQL Database

- **Category:** Information Disclosure

- **Description:** Weak authorization allows sensitive information exposure.
- **Mitigation:** None provided.

3.5 Denial of Service Threats

Resource Exhaustion (DOS)

- **Category:** DoS
- **Description:** System resources may be overloaded, causing service outage.
- **Mitigation:** None provided.

3.6 Elevation of Privilege Threats

1. Elevation Using Impersonation

- **Category:** EoP
- **Description:** Web server may impersonate human user contexts to gain additional privileges.
- **Mitigation:** None documented.

---

4. Overall Risk Assessment

Based strictly on the OWASP Threat Report

STRIDE Category	Number of Threats	Risk Level	Notes
Spoofing	2	High	Authentication gaps

Tampering	4+	High	XSS, log integrity issues
Repudiation	3	Medium–High	Insufficient auditing & log protection
Information Disclosure	3+	High	Weak access controls
Denial of Service	1+	High	Resource starvation possible
Elevation of Privilege	1	High	Impersonation risk

Even though many threats are labeled “Mitigation Implemented,” several show **no actual mitigation details**, implying incomplete closure.

---

5. Recommendations

5.1 Authentication & Spoofing

- Enforce mutual TLS between services.
- Authenticate all data stores using strong service-to-service identity (certificates or tokens).
- Apply strict IAM and role separation.

5.2 Input Validation & XSS Protection

- Implement centralized input validation pipeline.
- Enforce output encoding across web server components.
- Regular security code reviews and automated scanning (SAST/DAST).

5.3 Logging & Repudiation

- Harden logs using append-only storage.
- Use a SIEM with immutable log storage (e.g., Azure Log Analytics, Splunk).
- Configure logs to capture authentication failures, access patterns, and admin-level actions.

5.4 Access Control & Data Protection

- Apply attribute-based access control (ABAC).
- Encrypt DB records using key vault-managed keys.
- Mask sensitive fields even for privileged users.

5.5 DoS Hardening

- Introduce rate-limiting and load balancing.
- Use WAF and anti-DDoS services (e.g., Cloudflare, Azure DDoS Protection).
- Implement resource timeout and circuit breakers.

5.6 EoP Prevention

- Implement strict privilege boundaries between web server and database layers.
- Use impersonation controls and session isolation.
- Enforce principle of least privilege (PoLP).

6. Conclusion

The threat model demonstrates that your SQL-centric architecture has undergone initial threat analysis, but several threats lack detailed mitigation justification. Strengthening authentication, validating inputs, improving logging, and securing data access paths remain critical.

This report transforms the raw threat-modeling output into an actionable, security-focused document aligned with STRIDE best practices.