Threat Modelling Report – NXG Supply System

1. Executive Summary

This report documents a threat modelling exercise conducted on a Web-to-Database architecture responsible for handling user interactions and backend data storage. The system supports user authentication, transactional processing, and the storage of sensitive information, making it a critical component of the application's security posture.

Threat modelling was performed to identify potential security weaknesses early in the design phase, allowing risks to be addressed before deployment. Identifying threats at this stage reduces the likelihood of data breaches, service disruption, and costly remediation efforts later in the system lifecycle.

The assessment was conducted by analysing the system from an attacker's perspective. Data Flow Diagrams (DFDs) were used to visualise system components, data flows, and trust boundaries, while the STRIDE framework was applied to systematically categorise potential threats across the architecture.

The most critical risks identified include:

1. Unauthorized Data Access : The possibility of an attacker gaining direct or indirect access to sensitive data stored in the database.

2. Identity Spoofing : The risk of a malicious actor impersonating a trusted user or system component to intercept or manipulate data.

3. Audit and Log Tampering : The potential for attackers to alter or delete system logs, hindering incident detection and forensic investigations.

Overall, the system demonstrates a reasonable baseline security design; however, several areas require improvement to strengthen trust boundaries and protect sensitive assets. Implementing the recommended mitigations would significantly enhance the system's security posture and reduce overall business risk.

2. System Overview

The system follows a **three-tier architecture**, consisting of a web interface, an application server, and a central SQL database. Users interact with the system through the web layer, where requests are processed by backend services and relevant data is retrieved or stored in the database.

Users of the system include:

- **Authorized Customers:** External users who access the application to view or update personal information.

- **Administrative Staff:** Internal users responsible for managing backend operations, system configuration, and database maintenance.

The system processes and stores the following data:

- **User Credentials:** Authentication data used to verify user identities.
- **Transactional Records:** Logs of user actions and system operations.
- **System Logs:** Records of application and server events used for monitoring and incident response.

This data is considered sensitive due to its potential impact if compromised. Unauthorized access could result in identity theft, data manipulation, regulatory non-compliance, and reputational damage. Maintaining the confidentiality, integrity, and availability of this information is therefore essential to the organisation's operations and trustworthiness.

3. Scope and Assumptions

Scope

Included:

- Security of data flows between the **User**, **Backend Web Server**, and **SQL Database**
- Credential storage and authentication data handling
- Integrity and protection of application and audit logs
- Security of backend-to-database connections

Excluded:

- Physical security of data centre infrastructure
- Reliability and security controls of third-party Internet Service Providers (ISPs)
- Client-side hardware and endpoint device security

Assumptions

- The application is internet-facing and accessible via standard web browsers
- Authentication is based on a username and password mechanism
- The environment does not implement a full Zero Trust architecture and relies on internal trust boundaries that require additional hardening

4. Threat Modelling Methodology

Rationale

A structured threat modelling methodology was adopted to ensure a repeatable and comprehensive identification of potential security risks. This approach helps prevent reactive security practices by identifying attack vectors early in the design phase.

Frameworks and Techniques

- **STRIDE Framework:** Used to categorise threats into Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.
- **Data Flow Diagrams (DFDs):** Used to visualise system components, data movement, and trust boundaries where threats are most likely to occur.

Tools Used

- **Microsoft Threat Modeling Tool:** Used for detailed technical analysis of backend services and database interactions.
- **OWASP Threat Dragon:** Used for high-level architectural modelling and collaborative threat identification.

5. Data Flow Diagram (DFD) Analysis

Analysis of the Data Flow Diagram identified the primary trust boundary between the **external internet** and the **Backend Web Server**, representing the highest exposure to unauthenticated traffic.

A secondary critical trust boundary exists between the **Backend Web Server** and the **SQL Database**. This zone was classified as high risk due to the potential for lateral movement if the web server is compromised. The analysis focused on securing this internal communication channel through encryption and strong authentication to prevent database spoofing, unauthorised access, and indirect attack techniques.

6. Threat Identification

Using the STRIDE framework, several credible attack scenarios were identified:

- **Spoofing:** An attacker impersonating a legitimate database endpoint to intercept or manipulate data.
- **Tampering:** Injection of malicious payloads into application logs to exploit administrative interfaces.
- **Repudiation:** Deletion or manipulation of audit logs to deny malicious actions.
- **Information Disclosure:** Inadequate access controls allowing unauthorised reading of sensitive database records.

7. Risk Assessment

Threat	STRIDE Category	Impact	Likelihood	Risk Level
Database Spoofing	Spoofing	High	Medium	High
Log Injection	Tampering	Medium	Medium	Medium
Audit Log Deletion	Repudiation	High	Low	Medium
Unauthorised Data Access	Information Disclosure	High	High	Critical

Risk levels were determined based on potential business impact and the likelihood of exploitation in an internet-facing environment.

8. Mitigations and Remediations

To address the identified risks, the following mitigations are recommended:

- **Database Spoofing:** Implement **Mutual TLS (mTLS)** to ensure mutual authentication between the application server and database.
- **Log Tampering:** Apply **input canonicalisation and validation** before log entries are written to prevent injection-based attacks.
- **Unauthorised Data Access:** Enforce **Role-Based Access Control (RBAC)** and **row-level security** to ensure least-privilege access.
- **Data Exposure from Stolen Files:** Use **Transparent Data Encryption (TDE)** to protect data at rest in the event of database file compromise.

9. Tool Evaluation and Comparison

OWASP Threat Dragon

Strengths:

- Strong visual modelling capabilities
- Web-based and collaborative
- Beginner-friendly interface

Weaknesses:

- Limited automated threat generation
- Less suitable for deep technical analysis

Microsoft Threat Modeling Tool

Strengths:

- Automated threat generation based on component stencils
- Comprehensive threat libraries aligned with enterprise environments

Weaknesses:

- Steeper learning curve
- Windows-only platform

Recommendation:
In an enterprise environment, the Microsoft Threat Modeling Tool would be preferred for final technical validation due to its comprehensive and automated threat identification capabilities.

10. Limitations and Future Improvements

This threat model was conducted at the **design level** and does not validate implementation-specific issues within source code or deployed systems.

Future work would include:

- Dynamic testing and penetration testing
- Expansion to include cloud infrastructure components (e.g., AWS or Azure)
- Re-evaluation of trust boundaries in a Zero Trust architecture

11. Conclusion

This threat modelling exercise identified critical weaknesses related to database trust, access control, and logging integrity. By implementing mitigations such as mTLS, RBAC, and TDE, the system's defence-in-depth posture is significantly improved. Threat modelling has enabled security considerations to be embedded into the system design rather than addressed reactively.

12. Final Statement

This threat modelling exercise was conducted in a controlled lab environment for educational purposes as part of a cybersecurity portfolio.