Threat Modeling Report

Created on 2/4/2026 6:23:26 PM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	11
Total	11
Total Migrated	0

Diagram: Diagram 1

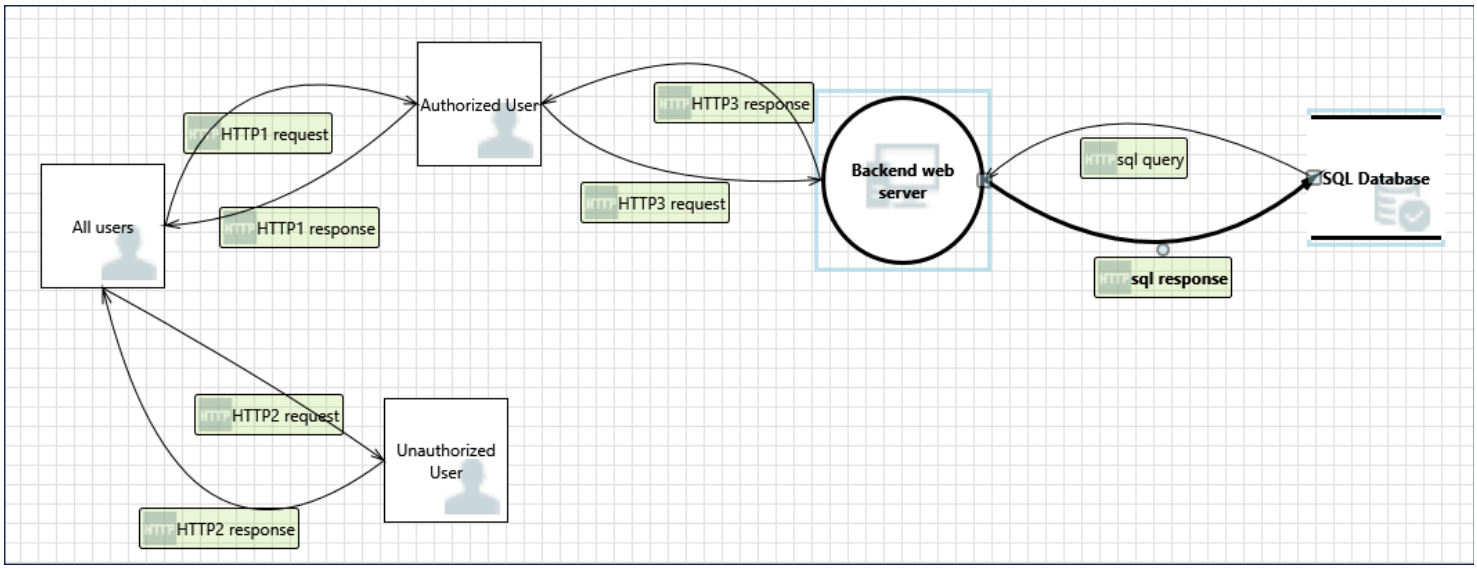
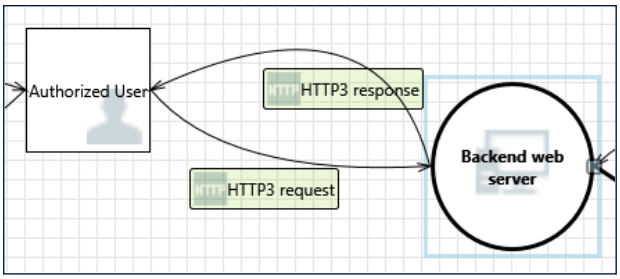


Diagram 1 Diagram Summary:

Not Started	0
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	11
Total	11
Total Migrated	0

Interaction: HTTP3 request



1. Spoofing the Authorized User External Entity [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: Authorized User may be spoofed by an attacker and this may lead to unauthorized access to Backend web server. Consider using a standard authentication mechanism to identify the external entity.
Justification: Ensure the use of a standard authentication API

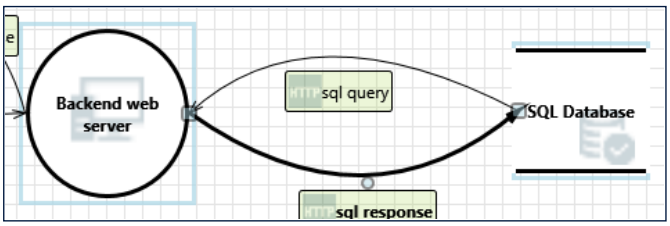
2. Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: The web server 'Backend web server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>

3. Elevation Using Impersonation [State: Mitigation Implemented] [Priority: High]

Category: Elevation Of Privilege
Description: Backend web server may be able to impersonate the context of Authorized User in order to gain additional privilege.
Justification: <no mitigation provided>

Interaction: sql query



4. Spoofing of Source Data Store SQL Database [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: SQL Database may be spoofed by an attacker and this may lead to incorrect data delivered to Backend web server. Consider using a standard authentication mechanism to identify the source data store.
Justification: <no mitigation provided>

5. Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: The web server 'Backend web server' could be a subject to a cross-site scripting attack because it does not sanitize untrusted input.
Justification: <no mitigation provided>

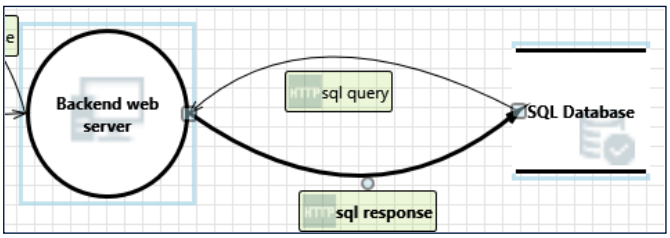
6. Persistent Cross Site Scripting [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: The web server 'Backend web server' could be a subject to a persistent cross-site scripting attack because it does not sanitize data store 'SQL Database' inputs and output.
Justification: <no mitigation provided>

7. Weak Access Control for a Resource [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. Review authorization settings.
Justification: <no mitigation provided>

Interaction: sql response



8. Authorization Bypass [State: Mitigation Implemented] [Priority: High]

Category: Information Disclosure
Description: Can you access SQL Database and bypass the permissions for the object? For example by editing the files directly with a hex editor, or reaching it via filesharing? Ensure that your program is the only one that can access the data, and that all other subjects have to use your interface.
Justification: <no mitigation provided>

9. Spoofing of Destination Data Store SQL Database [State: Mitigation Implemented] [Priority: High]

Category: Spoofing
Description: SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.
Justification: <no mitigation provided>

10. Potential SQL Injection Vulnerability for SQL Database [State: Mitigation Implemented] [Priority: High]

Category: Tampering
Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.
Justification: <no mitigation provided>

11. Potential Excessive Resource Consumption for Backend web server or SQL Database [State: Mitigation Implemented] [Priority: High]

Category: Denial Of Service
Description: Does Backend web server or SQL Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification: <no mitigation provided>