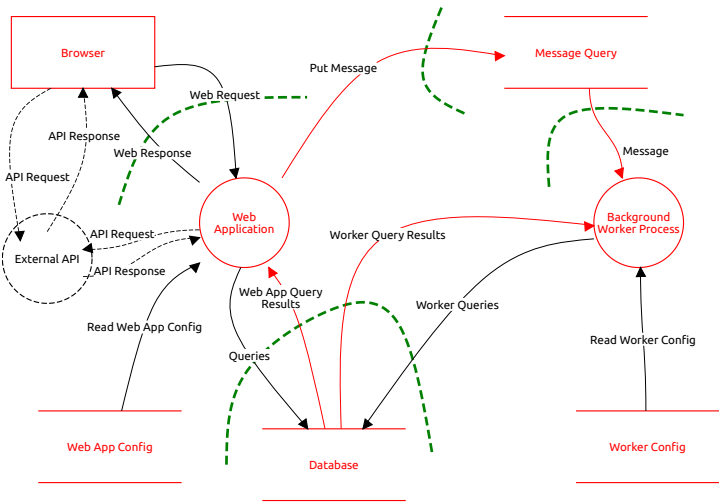


Threat model report for Front End app

Owner:                      Reviewer:                      Contributors:

High level system description

all



Browser (External Actor)

Generic spoofing threat  
*Spoofing, Open, Medium Severity*

**Description:**  
A generic spoofing threat  
**Mitigation:**

Generic repudiation threat  
*Repudiation, Open, Medium Severity*

**Description:**  
A generic repudiation threat  
**Mitigation:**

Web App Config (Data Store)

Credentials should be encrypted  
*Information disclosure, Open, High Severity*

**Description:**  
The Web Application Config stores credentials used by the Web App to access the message queue. These could be stolen by an attacker and used to read confidential data or place poison message on the queue.  
**Mitigation:**  
The Message Queue credentials should be encrypted.

Generic DoS threat  
*Denial of service, Open, Medium Severity*

**Description:**  
A generic DoS threat

**Mitigation:**

## Database (Data Store)

## Credential theft

*Information disclosure, Open, Medium Severity***Description:**

An attacker could obtain the DB credentials and use them to make unauthorised queries.

**Mitigation:**

Use a firewall to restrict access to the DB to only the Background Worker IP address.

## Unauthorised access

*Information disclosure, Open, Medium Severity***Description:**

An attacker could make a query call on the DB,

**Mitigation:**

Require all queries to be authenticated.

## Worker Config (Data Store)

## Accessing DB credentials

*Information disclosure, Open, High Severity***Description:**

The Background Worker configuration stores the credentials used by the worker to access the DB. An attacker could compromise the Background Worker and get access to the DB credentials.

**Mitigation:**

Encrypt the DB credentials in the configuration file. Expire and replace the DB credentials regularly.

## Message Query (Data Store)

## Message secrecy

*Information disclosure, Open, Low Severity***Description:**

The data flow between the Web Application and the Background Worker is not point-to-point and therefore end-to-end secrecy cannot be provided at the transport layer. Messages could be read by an attacker at rest in the Message Queue.

**Mitigation:**

Use message level encryption for high sensitivity data (e.g. security tokens) in messages.

## Message tampering

*Tampering, Open, Medium Severity***Description:**

Messages on the queue could be tampered with, causing incorrect processing by the Background Worker.

**Mitigation:**

Sign all queue messages at the Web Server. Validate the message signature at the Background Worker and reject any message with a missing or invalid signature. Log any failed messages.

## Fake messages could be placed on the queue

*Spoofing, Open, High Severity***Description:**

An attacker could put a fake message on queue, causing the Background Worker to do incorrect processing.

**Mitigation:**

Restrict access to the queue to the IP addresses of the Web Server and Background Worker. Implement authentication on the queue endpoint.

## Web Application (Process)

## Generic spoofing threat

*Spoofing, Open, Medium Severity*

**Description:**  
A generic spoofing threat  
**Mitigation:**

Generic tampering threat  
*Tampering, Open, Medium Severity*

**Description:**  
A generic tampering threat  
**Mitigation:**

Generic repudiation threat  
*Repudiation, Open, Medium Severity*

**Description:**  
A generic repudiation threat  
**Mitigation:**

Generic information disclosure threat  
*Information disclosure, Open, Medium Severity*

**Description:**  
A generic information disclosure threat  
**Mitigation:**

Generic DoS threat  
*Denial of service, Open, Medium Severity*

**Description:**  
A generic DoS threat  
**Mitigation:**

Generic elevation threat  
*Elevation of privilege, Open, Medium Severity*

**Description:**  
A generic elevation threat  
**Mitigation:**

#### Background Worker Process (Process)

Poison messages 1  
*Denial of service, Open, Medium Severity*

**Description:**  
An attacker could generate a malicious message that the Background Worker cannot process.  
**Mitigation:**  
Implement a poison message queue where messages are placed after a fixed number of retries.

Poison messages 2  
*Denial of service, Open, Medium Severity*

**Description:**  
An attacker could generate a malicious message that the Background Worker cannot process.  
**Mitigation:**  
Validate the content of all messages, before processing. Reject any message that have invalid content and log the rejection. Do not log the malicious content - instead log a description of the error.

#### Web Request (Data Flow)

*No threats listed.*

#### Web Response (Data Flow)

*No threats listed.*

## Put Message (Data Flow)

Data flow should use HTTP/S

*Information disclosure, Open, High Severity*

**Description:**

These requests are made over the public internet and could be intercepted by an attacker.

**Mitigation:**

The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

## Read Web App Config (Data Flow)

*No threats listed.*

## Queries (Data Flow)

*No threats listed.*

## Web App Query Results (Data Flow)

Server Name and Type

*Information disclosure, Open, Medium Severity*

**Description:**

The server responds with the the proxy library and version

**Mitigation:**

Remove from response as it's not necessary or encrypt.

## Worker Query Results (Data Flow)

Man in the middle attack

*Information disclosure, Open, Low Severity*

**Description:**

An attacker could intercept the DB queries in transit and obtain sensitive information, such as DB credentials, query parameters or query results (is unlikely since the data flow is over a private network).

**Mitigation:**

Enforce an encrypted connection at the DB server

## Message (Data Flow)

Data flow should use HTTP/S

*Information disclosure, Open, High Severity*

**Description:**

These requests are made over the public internet and could be intercepted by an attacker.

**Mitigation:**

The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

## Worker Queries (Data Flow)

*No threats listed.*

## Read Worker Config (Data Flow)

*No threats listed.*

## External API (out of scope Process)

**Out of scope reason:**

This API is not ours and thus out of scope.

API Request (out of scope Data Flow)
<b>Out of scope reason:</b> This API is not ours and thus outof scope.
API Response (out of scope Data Flow)
<b>Out of scope reason:</b> This API is not ours and thus outof scope.
API Response (out of scope Data Flow)
<b>Out of scope reason:</b> This API is not ours and thus outof scope.
API Request (out of scope Data Flow)
<b>Out of scope reason:</b> This API is not ours and thus outof scope.