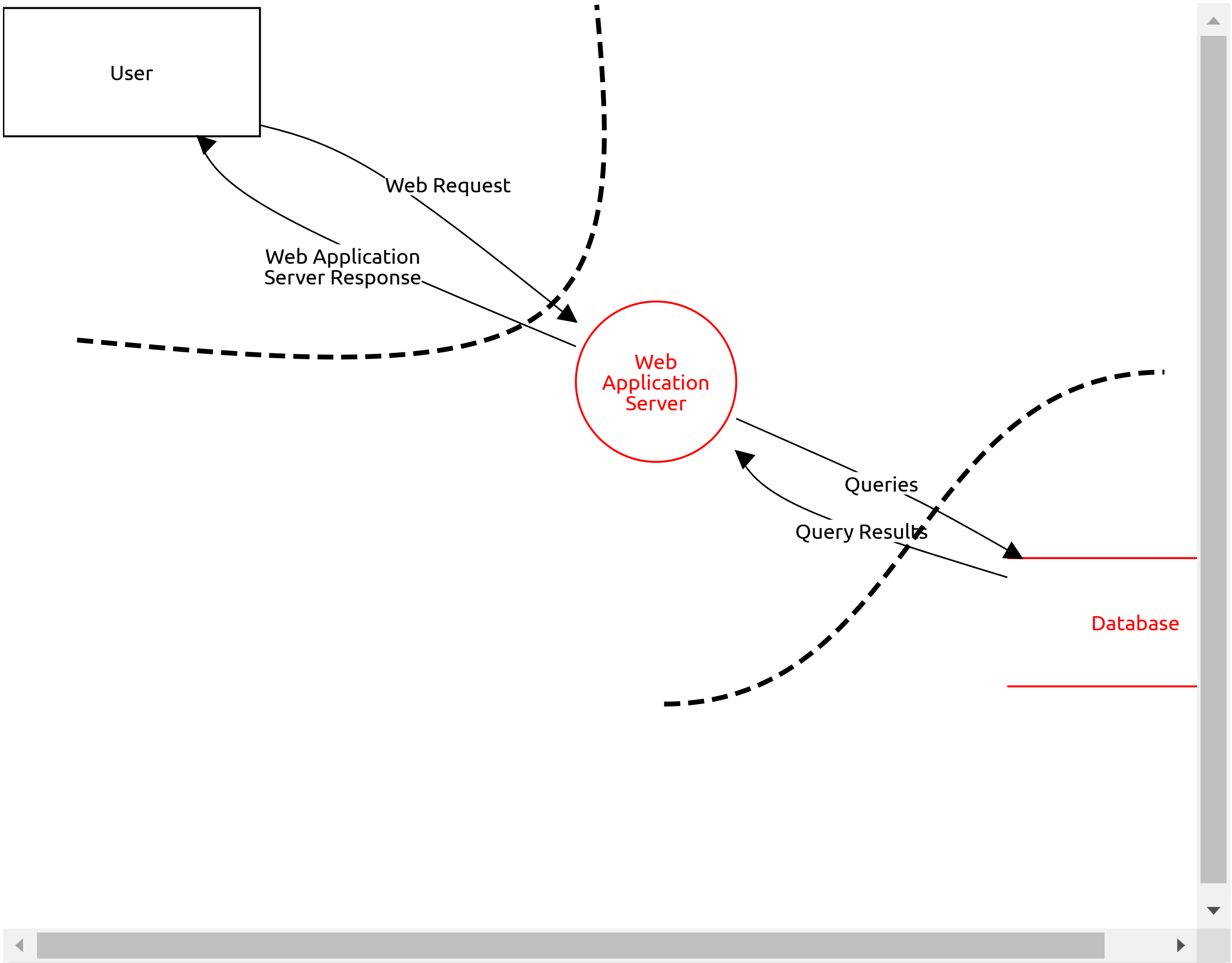
Threat model report for Restaurant Rater

Owner:
Faizan Hussain
Reviewer:
Faizan Hussain
Contributors:

High level system description

Restaurant Rater is a web-based application made to allow users to add their favorite restaurants, rate, and review them. Users are allowed to create an account, login/logout, add restaurants, review and rate restaurants, and edit/delete restaurants they created.

Threat Model



User (External Actor)

Description:

No threats listed.

Database (Data Store)

Description:

SQL Injection

Information disclosure, Mitigated, High Priority

Description:
SQL Injection can lead to confidential information being leaked such as log in credentials.

Mitigation:
Parameterized Queries and hashing passwords added to the database

SQL Injection

Tampering, Mitigated, High Priority

Description:
SQL Injections can tamper and manipulate the data stored in the database.

Mitigation:
Parameterized Queries

Credential theft

Information disclosure, Open, Medium Priority

Description:
An attacker could obtain the DB credentials and use them to make unauthorized queries.

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

Web Application Server (Process)

Description:

DDoS

Denial of service, Open, High Priority

Description:
Without API Rate limiting, the API endpoints are vulnerable to DDoS Attacks, causing increased network traffic. This vulnerability has been exploited through a python script sending 100s of review requests to the review API.

Mitigation:
Setting API Rate limiters either based off an IP address or the user's account.

Unauthorized API Calls

Elevation of privilege, Open, High Priority

Description:
Some APIs do not require a JWT to make an API call, therefore allowing those APIs to be called by users who are not even logged in.

Mitigation:
Adding authorization middleware to the appropriate API endpoints. Middleware is already developed, just needs to be called in the API code.

Fake names can be used for leaving reviews

Spoofing, Open, Low Priority

Description:
Due to the API in charge of the reviews for restaurants not having proper authorization required, adversaries can send requests to the API endpoint with any name in the header. Therefore, allowing one to act as someone else leaving a review for a restaurant instead of themselves.

Traditionally, the name of the user leaving the review is taken directly from the account information of the currently logged on user.

Mitigation:
Requiring authorization for the API in charge of leaving reviews to ensure that the name can only be yielded from the logged in user.

Queries (Data Flow)

Description:

Man in the middle attack

Information disclosure, Mitigated, High Priority

Description:
An attacker could intercept the DB queries in transit and obtain sensitive information, such as DB credentials, query parameters or query results.

Mitigation:
Enforce an encrypted connection at the DB server

Web Request (Data Flow)

Description:

Data Flow Should use HTTP/s

Information disclosure, Mitigated, High Priority

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

Mitigation:

Query Results (Data Flow)

Description:

No threats listed.

Web Application Server Response (Data Flow)

Description:

No threats listed.