M2: Data Flow Diagram

A black background with white text

Description automatically generated

Skylar Siggard

College of Engineering and Technology

Utah Valley University

September 11, 2024

IT 6780 Secure Coding

Kodey S. Crandall, PH.D

# Abstract

This report analyzes the security of The Recipe Hub, a web application built with Next.js, Clerk for authentication, and SQLite for data storage. It presents Data Flow Diagrams to map key processes, such as user authentication, recipe submission, and email communications. The goal is to ensure that The Recipe Hub is secure and resilient against common vulnerabilities.

Table of Contents

[Abstract 2](#_Toc176984211)

[1 Introduction 4](#_Toc176984212)

[2 Background 4](#_Toc176984213)

[2.1 Level 0 DFD 4](#_Toc176984214)

[2.2 Level 1 DFD 5](#_Toc176984215)

[3 Threat Modeling 5](#_Toc176984216)

[4 Conclusions 7](#_Toc176984217)

# 1 Introduction

To protect user data, system integrity, and ensure a secure architecture we need to understand how our application works. We can start to understand key areas by creating a Data Flow Diagram.

# 2 Background

The Recipe Hub is a web application designed to provide users with a platform to browse, submit, manage, and share recipes. Built using Next.js for the frontend, Clerk for authentication, and SQLite for data management, the application offers essential features like recipe browsing, rating, commenting, and social sharing.

## 2.1 Level 0 DFD

Level 0 simplifies complex systems, help everyone understand key interactions and boundaries. This foundational diagram is crucial for establishing context and serves as a basis for more detailed DFDs, while also supporting communication between teams and aiding in identifying security risks at the system’s entry and exit points.

Users interact with Recipe Hub application by visiting <https://it-6780-repo.vercel.app>. Once authenticated by login in they can view their recipe list and create new recipes. The view they see is handled by the Next.js framework. They can request data if they are authenticated user. Data is linked to each account and can be shared via email to other users. Once shared the other use may add this recipe to their account making the recipe connected to their account.

## 2.2 Level 1 DFD

Level 1 identifies key processes, data stores, and interactions that occur internally, giving a clearer view of how the system works. This level of detail is important for understanding the individual functions and for pinpointing potential vulnerabilities, as well as supporting more in-depth analysis for development, troubleshooting, and security.

Users interact with Recipe Hub application by visiting <https://it-6780-repo.vercel.app>. During the registration phase Next.js communicates to Clerk Auth via an API. Clerk is then told to create a token to provide Firebase and is stored by Firebase. Clerk with the help of the stored token within Firebase the user receives a session and is authenticated. With this session created the user may create, rate, share recipes. These recipes are stored on an SQLite database. All data is connected to individual accounts and can only retrieve recipes associated with their accounts. Recipes are connected to the accounts if they are the creator of the recipe or if it was shared to them. Data retrieved from SQLite Database will be placed into an email and sent via Next.js Resender.

# 3 Threat Modeling

Threat modeling is important because it helps identify, assess, and mitigate potential security risks in a system before they are exploited. It helps prioritize risks based on their potential impact and likelihood, ensuring that resources are focused on the most critical threats.

**Clerk Authentication**

* Threat ID: T1
* Threat Description: Attackers attempts to bypass Clerk Authentication by intercepting session tokens.
* Threat Type: Spoofing.
* Impact: Unauthorized access to user’s accounts.
* Likelihood: Medium.
* Mitigation: Use HTTPS for all traffic, securely store session tokens, set short-lived tokens, and enable multi authentication.

**Next.js API Routes**

* Threat ID: T2
* Threat Description: Attacker uses a poorly secure API endpoint to manipulate data in the SQLite Database.
* Threat Type: Tampering.
* Impact: Unauthorized modification of recipe data.
* Likelihood: High.
* Mitigation: Implement input validation (such as ZOD) and authentication checks on all API routes.

**SQLite Database**

* Threat ID: T3
* Threat Description: SQL injection in recipe submission forms allows attackers to extract or modify data in SQLite.
* Threat Type: Information Disclosure/ Tempering.
* Impact: Database compromise.
* Likelihood: High.
* Mitigation: Use parameterized queries and ORM (e.g. Drizzle ORM) to prevent SQL injection.

**Email Content Injection**

* Threat ID: T4
* Threat Description: An attacker exploits weaknesses in how email content is generated, inserting malicious content such as links to phishing sites or malware.
* Threat Type: Tampering.
* Impact: Users could be tricked into clicking harmful links, leading to malware infection or data theft.
* Likelihood: Medium.
* Mitigation: Ensure proper sanitization of all user’s inputs that is included in email content. Validate and encode any dynamic content before rendering it in the email.

# 4 Conclusions

The Data Flow Diagram for The Recipe Hub provides a clear view of how data flows through the system. With this data we can strive to secure this application.