Abuse Use Cases for Ticketmaster-Clone Project

Abuse Use Case 1: Price Manipulation in Cart

Description

In this scenario, a malicious user tries to manipulate the price of tickets in the cart by intercepting and altering the web requests sent from the client to the server.

Countermeasures

1. Server-side Validation: Use Spring Boot's @Valid annotation for server-side validation of incoming request payloads to ensure correct pricing.
2. User Session Management: Utilize Spring Security for secure session management. Store session IDs securely in the database using PostgreSQL.
3. Use HTTPS: Configure Spring Boot to force HTTPS by redirecting HTTP traffic. This can be done using Spring Security's requiresChannel() method.

Abuse Use Case 2: Payment Infiltration

Description

A malicious user attempts to infiltrate the payment process to either steal sensitive information like credit card details or to manipulate the transaction.

Countermeasures

1. Use Stripe API: Implement Stripe's API in your Spring Boot backend to handle payments. Stripe SDK has built-in security features that offload the sensitive part of handling payments.
2. Server-Side Verification: In your Spring Boot application, use services to validate transaction information before sending it to Stripe.
3. HTTPS: As in the first case, configure Spring Boot to force HTTPS using Spring Security to ensure encrypted data communication.