PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	• Protect customer privacy by responsibly collecting, storing and processing account, message and payment data. • Deliver friction-free sign-up/login plus in-app messaging and seller-rating features that build user trust and community. • Complete purchases quickly through several payment options while meeting relevant financial-data regulations.
II. Define the technical scope	List of technologies used by the application: • *Application programming interface (API)* The API is the internet-facing entry point for every workflow—authentication, product search, messaging, ratings and checkout. If its authentication, input-validation or rate-limiting logic is weak, an attacker can steal data or alter transactions before deeper defenses (encryption, database controls) engage, making it the riskiest component to examine first (59 words). Additional critical technologies: the PKI stack that combines AES and RSA for transport-layer encryption, and the SQL database that stores listings, credentials and payment tokens.
III. Decompose application	Sample data flow diagram
IV. Threat analysis	• *SQL-injection attacks that exfiltrate or modify inventory and user tables* • *Credential-stuffing or brute-force logins that hijack buyer or seller accounts and their stored payment methods.*

V. Vulnerability analysis	• Dynamic SQL statements that do not use prepared queries, leaving the inventory-search process in the data-flow diagram open to injection • Insecure storage of private RSA keys (for example, hard-coding them in the mobile app), allowing decryption of traffic captured from man-in-the-middle attacks.
VI. Attack modeling	Sample attack tree diagram
VII. Risk analysis and impact	1. Enforce multi-factor authentication for all accounts to reduce the impact of credential reuse. 2. Adopt parameterized queries or an ORM layer and run automated SQL-injection tests in CI to eliminate unsafe database calls. 3. Place the API behind a Web Application Firewall/API gateway with strict rate limiting, input-validation rules and anomaly detection. 4. Store encryption keys in a hardware security module (HSM) and require TLS 1.3 with HSTS for every session to prevent key theft and eavesdropping.