Scenario

You're part of the growing security team at a company for sneaker enthusiasts and collectors. The business is preparing to launch a mobile app that makes it easy for their customers to buy and sell shoes.

You are performing a threat model of the application using the PASTA framework. You will go through each of the seven stages of the framework to identify security requirements for the new sneaker company app.

The application should seamlessly connect sellers and shoppers. It should be easy for users to sign-up, log in, and manage their accounts. Data privacy is a big concern for us. We want users to feel confident that we're being responsible with their information.

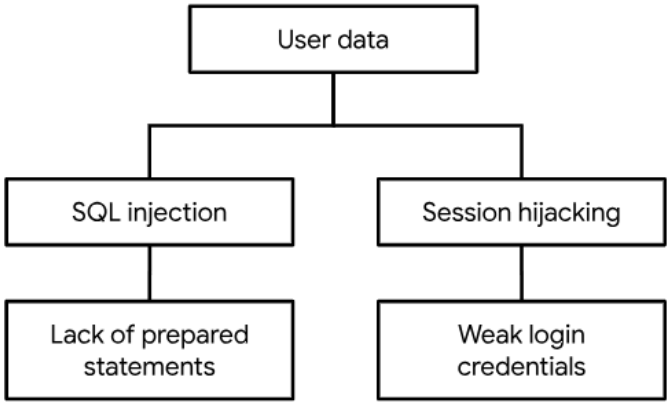
Buyers should be able to directly message sellers with questions. They should also have the ability to rate sellers to encourage good service. Sales should be clear and quick to process. Users should have several payment options for a smooth checkout process. Proper payment handling is really important because we want to avoid legal issues.

Data Flow Diagram

Note: This data flow diagram represents a single process. Data flow diagrams for an application like this are normally much more complex.

Sample Attack Tree

Note: Applications like this normally have large, complex attack trees with many branches.



PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	The app must securely process transactions through multiple payment options, ensuring compliance with industry regulations. It will handle significant back-end processing, including data storage, serving user data, transaction processing, and account creation. Compliance with relevant industry regulations, such as PCI DSS for payment processing and data protection laws, must be strictly adhered to.
II. Define the technical scope	List of technologies used by the application: ● *Application programming interface (API)* ● *Public key infrastructure (PKI)* ● *SHA-256* ● *SQL* In this case, the main focus would be on the APIs used for the application, as they define the functionality and integration with other systems. The choice of APIs becomes critical for implementing secure and efficient backend processes, handling transactions, and interacting with the database, ensuring both functionality and data security.
III. Decompose application	Sample data flow diagram (see above)
IV. Threat analysis	● SQL Injection ● Session Hijacking
V. Vulnerability analysis	● No prepared SQL Statements ● Broken API Token
VI. Attack modeling	Sample attack tree diagram (see above)
VII. Risk analysis and impact	● Input Validation ● Access Control/Principle of Least Privilege ● Data Encryption/Hashing ● Incident Response Procedures