PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	*Will the app process transactions?* Due to the nature of selling shoes, the application will process transactions, and offer several payment options for customers. The app will be doing a lot of back-end processing, as it is running an online shopping market. The database of products and PKI infrastructure are two of the most important backend components. The PCI DSS regulation needs to be strongly considered among the development team. Since the application deals with a lot of credit card payments/processing. The app also needs to abide by GDPR regulations, as it connects buyers and sellers all around the world, including Europe.
II. Define the technical scope	List of technologies used by the application: • *Application programming interface (API)* • *Public key infrastructure (PKI)* • *SHA-256* • *SQL* Out of these four technologies, securing the application's customer information database is the most important. Improper sanitization can allow malicious actors to use SQL injection attacks to modify, delete, and steal information from the database.
III. Decompose application	Sample data flow diagram
IV. Threat analysis	A potential internal threat could be the accidental leaking/sharing of critical information. This could happen if there are not proper standards and policies in place. Secondly, server-side request forgeries could be a potential threat if the development team has not been properly adhering to security guidelines.
V. Vulnerability analysis	The codebase could be insecure due to improper code security guidelines. This could allow for API infrastructure vulnerabilities, like a broken authentication system or unsecured configurations. The database could easily be susceptible to SQL injections if its

	platform fails to include proper sanitization procedures. The network could itself be vulnerable if there are improper password safety configurations, or outdated software. There is also always the possibility of a social engineering attack if the employees are not aware enough.
VI. Attack modeling	Sample attack tree diagram
VII. Risk analysis and impact	- Input validation can help ensure that user input meets system expectations, which can mitigate the chance of a malicious SQL query. - Input sanitization would greatly help mitigate the risks associated with having a database of customer information. - Having secure coding practices can help reduce the risk of weak/insecure API infrastructure. - Role based access control can help reduce risk by strengthening the principle of least privilege.