PASTA worksheet (My Responses)

Stages	Sneaker company
I. Define business and security objectives	Make 2-3 notes of specific business requirements that will be analyzed. ● ***1:*** *Based on the scenario and notes, the application must allow users to create and manage profiles through both internal registration and third-party logins (e.g., social media or Google accounts). The user experience should be easy, fast, and secure. Profile creation should include storing personal data such as usernames, contact information, and payment preferences.* ● ***2:*** *Based on the scenario and notes, the app must process financial transactions in a secure and legally compliant manner. This includes support for credit card payments and digital wallets. The system must ensure that sensitive payment information is encrypted and follows PCI-DSS (Payment Card Industry Data Security Standard) guidelines.* ● ***3:*** *Based on the scenario and notes, the application must prioritize data privacy and trust. Users should be confident that their information (login credentials, messages, and payment history) is secure. The business also wants to ensure smooth, direct messaging between buyers and sellers and allow users to leave seller ratings to promote a trustworthy community.*
II. Define the technical scope	List of technologies used by the application: ● ***SQL:*** *Manages user profiles, product listings, and transaction data in the database.* ● ***APIs (Application Programming Interfaces):*** *Used for handling requests between users, sellers, payment processors, and third-party services.* ● ***AES Encryption:*** *Secures data-in-transit and sensitive fields like credit card numbers.* ● ***SHA-256:*** *Hashes passwords and other private data.* ● ***PKI (Public Key Infrastructure):*** *Supports encryption of communications using RSA and AES.*

	Explanation: *APIs are the most critical area to evaluate first due to their central role in connecting the mobile app to external services such as payment gateways, login providers, and database queries. Because APIs handle a lot of sensitive user and financial data, they introduce significant risks if not properly secured. For example, API's can all expose the system to threats and the large attack surface of APIs makes them more vulnerable compared to internal encryption or database systems, which operate behind controlled access points. Meanhwile, SQL is also a critical component to prioritize due to its direct interaction with the backend database that stores user data, product listings, and transaction records. If SQL queries are not properly protected using prepared statements or parameterized queries, attackers could exploit input fields through SQL injection. This could allow unauthorized access to sensitive information such as user credentials or payment details.*
III. Decompose application	Sample data flow diagram **Data Flow Breakdown (based on diagram):** 1. **User Input:** A user searches for shoes using a search bar. 2. **Search Processing:** The app receives the input and sends a query to retrieve matching inventory items. 3. **Inventory Listings:** The search results (product titles, descriptions, prices) are pulled from the database using SQL queries. 4. **Database Communication:** The SQL database returns relevant sneaker listings to the user. **Security Implications based on Scenario and Notes:** Since user input directly affects SQL queries, this layer must have strong input validation and use of **prepared statements** to prevent injection attacks. Communication between the client and the server must be encrypted using **AES** (implemented via **PKI**) to protect sensitive session data and search behavior. Even though this seems like a simple process, it's a major interaction point between the user and the backend, making it a potential entry point for attackers.
IV. Threat analysis	List **2 types of threats** in the PASTA worksheet that are risks to the information being handled by the application. ● ***SQL Injection:*** *This occurs when a user sends malicious input that is directly processed by the database. For example, an attacker could input ' OR '1'='1 into the search bar to retrieve all listings, or worse, access unauthorized*

	data. Without input sanitization or prepared statements, the backend is vulnerable to this threat. ● ***Session Hijacking:*** *If a user's session token is stolen (e.g., via an unsecured connection or XSS attack), the attacker can impersonate that user. This is especially dangerous if the user is an admin or seller. The threat becomes worse if sessions do not expire quickly or tokens are not rotated.*
V. Vulnerability analysis	List **2 vulnerabilities** in the PASTA worksheet that could be exploited. ● ***Not enough correct Prepared Statements:*** *Dynamic SQL queries that do not use parameterized queries are vulnerable to SQL injection. For example, SELECT * FROM shoes WHERE name = ' " + userInput + " ' allows attackers to inject malicious SQL. This vulnerability could expose the entire sneaker listing or even customer data if exploited.* ● ***Broken API Token(s) Management and Applications:*** *APIs may issue tokens that do not expire or are easy to guess. If the app doesn't verify the token correctly or uses the same token for a long time, attackers can intercept and reuse them, leading to unauthorized access. Missing token rotation, weak token structure, or lack of session expiration increases the attack surface.*
VI. Attack modeling	Sample attack tree diagram **Sample Attack Tree (Breakdown with the notes):** **User Data** -> **SQL Injection** -> Caused by lack of prepared statements **User Data** -> **Session Hijacking** -> Enabled by weak login credentials or unsecured sessions **Attack Scenario Example and explanation of the breakdown:** An attacker performs reconnaissance by analyzing the app's API responses. They find an endpoint that processes product searches and test various payloads. After discovering the backend doesn't use prepared statements, they inject malicious SQL to retrieve all user emails. Separately, the attacker captures a session token during a login process over a poorly configured HTTPS setup, gaining access to the victim's account.
VII. Risk analysis and impact	List **4 security controls** that can reduce risk. • ***Use SHA-256 for Password Hashing:*** *Based on the*

	scenario and notes, all user passwords must be stored using SHA-256 hashing with a salt to prevent brute-force and rainbow table attacks in case of database compromise. • ***Incident Response Procedures:*** *Based on the scenario and notes, a documented incident response plan should be in place. This includes identifying threats, containing damage, notifying users, and learning from the breach to harden defenses. Timely responses limit the scope of damage and regulatory consequences.* • ***Enforce Strong Password and Authentication Policies:*** *Based on the scenario and notes, to improve security, implement a minimum password complexity, encourage multi-factor authentication (MFA), and lock accounts after multiple failed attempts to reduce the risk of unauthorized access.* • ***Principle of Least Privilege:*** *Based on the scenario and notes, implementing the principle of least privilege ensures users, services, and processes only have access to the data and features they absolutely need. For example, a seller should not be able to access payment logs, and an API should only retrieve data relevant to the request. This minimizes the potential impact of a compromised account or service.*

NOTES:

Activity Overview

In this activity, you will practice using the Process of Attack Simulation and Threat Analysis (PASTA) threat model framework. You will determine whether a new shopping app is safe to launch.

Threat modeling is an important part of secure software development. Security teams typically perform threat models to identify vulnerabilities before malicious actors do. PASTA is a commonly used framework for assessing the risk profile of new applications.

Scenario

Review the following scenario. Then complete the step-by-step instructions.

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.

Step 2: Access supporting materials

The following supporting materials will help you complete this activity. Keep them open as you proceed to the next steps.

To use the supporting materials for this course item, click the following link and select *Use Template*.
Link to supporting materials:

- PASTA data flow diagram
- PASTA attack tree

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.
User <----> Product search process <---->Listings of current inventory <----> Database

Sample attack tree
Note: Applications like this normally have large, complex attack trees with many branches.
User data -> SQL injection -> Lack of prepared statements
User data -> Session hijacking -> Weak login credentials

Step 1: Identify the mobile app's business objectives
The main goal of Stage I of the PASTA framework is to understand why the application was developed and what it is expected to do.
Note: *Stage I typically requires gathering input from many individuals at a business.*
First, review the following description of why the sneaker company decided to develop this new app:
Description: Our 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.
In the **Stage 1** row of the **PASTA worksheet,** make **2-3 notes** of business objectives that you've identified from the description.

Step 2: Evaluate the apps components
In Stage II, the technological scope of the project is defined. Normally, the application development team is involved in this stage because they have the most knowledge about the code base and application logic. Your responsibility as a security professional would be to evaluate the application's architecture for security risks.
For example, the app will be exchanging and storing a lot of user data. These are some of the technologies that it uses:

- **Application programming interface (API):** An API is a set of rules that define how software components interact with each other. In application development, third-party APIs are commonly used to add functionality without having to program it from scratch.
- **Public key infrastructure (PKI):** PKI is an encryption framework that secures the exchange of online information. The mobile app uses a combination of symmetric and asymmetric encryption algorithms: AES and RSA. AES encryption is used to encrypt sensitive data, such as credit card information. RSA encryption is used to exchange keys between the app and a user's device.
- **SHA-256:** SHA-256 is a commonly used hash function that takes an input of any length and produces a digest of 256 bits. The sneaker app will use SHA-256 to protect sensitive user data, like passwords and credit card numbers.
- **Structured query language (SQL):** SQL is a programming language used to create,

interact with, and request information from a database. For example, the mobile app uses SQL to store information about the sneakers that are for sale, as well as the sellers who are selling them. It also uses SQL to access that data during a purchase.

Consider what you've learned about these technologies:

- *Which of these technologies would you evaluate first? How might they present risks from a security perspective?*

In the **Stage II** row of the **PASTA worksheet**, write **2-3 sentences** (40-60 words) that describe why you choose to prioritize that technology over the others.

Step 3: Review a data flow diagram

During Stage III of PASTA, the objective is to analyze how the application is handling information. Here, each process is broken down.

For example, one of the app's processes might be to allow buyers to search the database for shoes that are for sale.

Open the **PASTA data flow diagram** resource. Review the diagram and consider how the technologies you evaluated relate to protecting user data in this process.

Note: *Software developers usually have detailed data flow diagrams available for security teams to use and verify that information is being processed securely.*

Step 4: Use an attacker mindset to analyze potential threats

Stage IV is about identifying potential threats to the application. This includes threats to the technologies you listed in Stage II. It also concerns the processes of your data flow diagram from Stage III.

For example, the apps authentication system could be attacked with a virus. Authentication could also be attacked if a threat actor social engineers an employee.

In the **Stage IV** row of the PASTA worksheet, list **2 types** of threats that are risks to the information being handled by the sneaker company's app.

Pro tip: *Internal system logs that you will use as a security analyst are good sources of threat intel*

Step 5: List vulnerabilities that can be exploited by those threats

Stage V of PASTA is the vulnerability analysis. Here, you need to consider the attack surface of the technologies listed in Stage II.

For example, the app will use a payment system. The form used to collect credit card information might be vulnerable if it fails to encrypt data.

In **Stage V** of the **PASTA worksheet**, list **2 types** of vulnerabilities that could be exploited.

Pro tip: *Resources like the CVE® list and OWASP are useful for finding common software vulnerabilities.* ***Step 6: Map assets, threats, and vulnerabilities to an attack tree***

In Stage VI of PASTA, the information gathered in the previous two steps are used to build an attack tree.

Open the ***PASTA attack tree*** *resource. Review the diagram and consider how threat actors can potentially exploit these attack vectors.*

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

Step 7: Identify new security controls that can reduce risk

PASTA threat modeling is commonly used to reduce the likelihood of security risks. In Stage VII, the final goal is to implement defenses and safeguards that mitigate threats.

In **Stage VII** of the **PASTA worksheet**, list **4 security controls** that you have learned about that can reduce the chances of a security incident, like a data breach.