PASTA Worksheet

Stages

Sneaker company

I. Define business and security objectives

Make 2-3 notes of specific business requirements that will be analyzed.

- Seamlessly connect buyers and sellers
- Ensure user data privacy and security
- Facilitate secure and efficient transactions

Will the app process transactions? Yes Does it do a lot of back-end processing? Likely, for user authentication, messaging, and payment processing Are there industry regulations that need to be considered? Yes, likely PCI DSS for payment processing and data privacy regulations like GDPR or CCPA.

II. Define the technical scope

List of technologies used by the application:

- Application programming interface (API)
- Public key infrastructure (PKI)
- SHA-256
- SQL

Write 2-3 sentences (40-60 words) that describe why you choose to prioritize that technology over the others.

I would prioritize **SQL** as it directly interacts with sensitive user and transaction data. Vulnerabilities in SQL implementation could lead to data breaches, unauthorized access, or manipulation of critical information. Ensuring the security of the database and its queries is paramount for protecting user trust and preventing financial losses.

III. Decompose application

Refer to the provided data flow diagram

IV. Threat analysis

List 2 types of threats in the PASTA worksheet that are risks to the information being handled by the application.

- **External Threats:**

    - SQL Injection attacks targeting the database to steal user data or gain unauthorized access.
    - Man-in-the-middle attacks attempting to intercept sensitive data during transmission.

What are the internal threats?

- Unauthorized access by employees to sensitive customer or financial data
- Accidental data leaks due to misconfigurations or human error

What are the external threats?

- Attacks targeting vulnerabilities in the API to gain unauthorized access or disrupt services.
- Brute-force attacks against user accounts to gain access to their information and transactions.

V. Vulnerability analysis

List 2 vulnerabilities in the PASTA worksheet that could be exploited.

- **Inadequate input validation in the API:** Allowing injection attacks (e.g., SQL injection).
- **Weak or reused passwords:** Making user accounts susceptible to brute-force attacks.

Could there be things wrong with the codebase? Yes, coding errors could introduce vulnerabilities. Could there be weaknesses in the database? Yes, insecure configurations or outdated database software could be exploited. Could there be flaws in the network? Yes, weak network security could enable man-in-the-middle attacks.

VI. Attack modeling

Refer to the provided attack tree diagram

VII. Risk analysis and impact

List 4 security controls that you've learned about that can reduce risk.

1. **Input validation and sanitization** to prevent injection attacks.
2. **Strong password policies and multi-factor authentication** to protect user accounts.
3. **Encryption of data at rest and in transit** to ensure confidentiality.
4. **Regular security audits and penetration testing** to identify and address vulnerabilities proactively.

Root Node (Goal): Compromise User Data or Disrupt App Functionality

Level 1 (Attack Categories)

- **External Attacks**
- **Internal Threats**

Level 2 (Specific Threats)

- **External Attacks**
    - SQL Injection
    - Man-in-the-Middle Attack
- **Internal Threats**
    - Unauthorized Access by Employees
    - Accidental Data Leaks

Level 3 (Vulnerabilities)

- **External Attacks**
    - SQL Injection:
        - Inadequate input validation in the API
    - Man-in-the-Middle Attack:
        - Weak network security
- **Internal Threats**
    - Unauthorized Access by Employees:
        - Weak or reused passwords
    - Accidental Data Leaks:
        - Misconfigurations
        - Lack of proper data handling procedures

Explanation

- **Root Node:** This is the attacker's ultimate goal.
- **Level 1:** Broad categories of attack approaches (external vs. internal).
- **Level 2:** Specific types of attacks that could be used to achieve the goal.
- **Level 3:** Vulnerabilities that enable those specific attacks.

Remember: This is a simplified attack tree. A real-world scenario would likely have more branches and sub-branches, representing various attack paths and potential vulnerabilities.

Visual Representation:

You can create a visual attack tree using various tools, like:

- **Draw.io:** A free online diagramming tool
- **Microsoft Visio:** (If you have access to it)
- **Specialized threat modeling tools:** ThreatModeler, etc.