## PASTA worksheet

| **Stages** | **Sneaker company** |
| --- | --- |
| **I. Define business and security objectives** | Make 2-3 notes of specific business requirements that will be analyzed.   * The app will process financial transactions, so secure and fast payment handling is critical. * It performs back-end operations like login management, messaging between users, and seller ratings. * It must comply with industry regulations related to data privacy (e.g. POPIA) and payment processing (e.g. PCI DSS). |
| **II. Define the technical scope** | List of technologies used by the application:  Application programming interface (API)  Public key infrastructure (PKI)  SHA-256  SQL  I’d prioritize SQL first because it’s what the app uses to manage and pull important data, like user accounts and sales info. If the SQL queries aren’t handled properly, the system could be vulnerable to injection attacks. That could lead to data being exposed or manipulated by attackers. |
| **III. Decompose application** | [Sample data flow diagram](https://docs.google.com/presentation/d/1ol7y79popTFfNHM-90ES-H-i1Lpd0YNvPShxBlXozjg/template/preview?resourcekey=0-DZAkf7Vzh2PXsP-j3oXV-g) |
| **IV. Threat analysis** | Social engineering attack: Targeting employees or sellers to trick them into revealing sensitive info.  Malware Infection – Malicious software could compromise authentication or payment systems. |
| **V. Vulnerability analysis** | * **Codebase Vulnerability:**  Bugs or insecure coding practices could lead to issues like broken authentication or improper input validation, opening the door to exploits like SQL injection or logic flaws. * **Database Vulnerability:**  Weak query protections or poor configuration could expose the system to injection attacks, unauthorized access, or data leakage. |
| **VI. Attack modeling** | [Sample attack tree diagram](https://docs.google.com/presentation/d/1FmWLyHgmq9XQoVuMxOym2PHO8IuedCkan4moYnI-EJ0/template/preview?usp=sharing&resourcekey=0-zYPY7AhPJdcClXamlAfOag) |
| **VII. Risk analysis and impact** | **List 4 security controls that can reduce risk.**   1. **Parameterized SQL Queries**  Prevents SQL injection by safely handling user input in database queries. 2. **Multi-Factor Authentication (MFA)**  Adds an extra layer of protection to user login, reducing risks from stolen passwords and phishing. 3. **Secure API Authentication & Validation**  Ensures that third-party integrations are verified and hardened against tampering or abuse. 4. **Employee Security Awareness Training**  Helps prevent social engineering by educating staff about phishing, impersonation, and safe practices. |