## PASTA worksheet

|  |  |
| --- | --- |
| **Stages** | **Sneaker company** |
| **I. Define business and security objectives** | Make **2-3 notes** of specific business requirements that will be analyzed.   * *Will the app process transactions?* * *Does it do a lot of back-end processing?* * *Are there industry regulations that need to be considered?*   *Yes, the app will process transactions, offering multiple payment options for a smooth checkout experience, making proper payment handling a key requirement.*  *Yes, the app will likely require significant back-end processing to handle user data, transactions, secure messaging between buyers and sellers, and account management.*    *Yes, data privacy and payment processing will need to comply with industry regulations such as GDPR for data protection and PCI DSS for secure payment handling to avoid legal issues.* |
| **II. Define the technical scope** | List oftechnologies used by the application:   * *API* * *PKI* * *AES* * *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 the \*\*API\*\* because it is the main interface for data exchange between the app and external systems, making it a high-risk target for attacks like unauthorized access or data breaches. Ensuring the API is secure with proper authentication and encryption is critical to protect sensitive user data and transaction details. |
| **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** | List **2 types of threats** in the PASTA worksheet that are risks to the information being handled by the application.   * *What are the internal threats?* * *What are the external threats?*   *1. \*\*Insider Misuse\*\*: Employees with access to sensitive data, such as user information or payment details, could misuse their privileges to leak or manipulate data for personal gain.*    *1. \*\*Phishing and Social Engineering\*\*: External attackers could use phishing or social engineering tactics to trick employees into revealing sensitive credentials or authentication details, allowing unauthorized access to the app’s systems.* |
| **V. Vulnerability analysis** | List **2 vulnerabilities** in the PASTA worksheet that could be exploited.   * *Could there be things wrong with the codebase?* * *Could there be weaknesses in the database?* * *Could there be flaws in the network?*   *1. \*\*Codebase Issues\*\*: There could be vulnerabilities in the codebase, such as insecure coding practices, inadequate input validation, or reliance on deprecated functions, which may lead to security exploits like buffer overflows or remote code execution.*  *2. \*\*Database Weaknesses\*\*: The database may have weaknesses, including insufficient encryption for stored sensitive data, overly permissive access controls, or failure to apply security patches promptly. These issues can result in unauthorized data access or data breaches.*  *3. \*\*Network Flaws\*\*: There could be flaws in the network configuration, such as unpatched firewalls, lack of network segmentation, or insufficiently secured APIs. These vulnerabilities can expose the application to man-in-the-middle attacks, data interception, or unauthorized access to backend systems.* |
| **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 you’ve learned about that can reduce risk.  ### Stage VII: Implement Defenses and Safeguards  1. \*\*API Security Controls\*\*: Implement strong authentication mechanisms, such as OAuth 2.0, and ensure all API endpoints are secured with proper access controls and encryption for data in transit.  2. \*\*Input Validation and Sanitization\*\*: Enforce rigorous input validation and sanitization to prevent injection attacks, such as SQL injection or cross-site scripting (XSS), by using parameterized queries and validating user input.  3. \*\*Database Encryption\*\*: Use encryption for sensitive data at rest and in transit to protect against unauthorized access. This includes implementing column-level encryption for sensitive fields, like credit card information and personally identifiable information (PII).  4. \*\*Regular Security Audits and Penetration Testing\*\*: Conduct regular security audits and penetration testing to identify and remediate vulnerabilities in the application, codebase, and network. This proactive approach helps ensure the security measures in place are effective and up to date. |