## 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?* * *Transaction Processing: Determine if the application will handle transactions. Understanding the volume, frequency, and nature of these transactions is crucial as it affects security protocols, encryption standards, and the robustness of the system to prevent financial data breaches.* * *Does it do a lot of back-end processing?* * *Backend Processing: Assess the extent of back-end processing the application conducts. This involves understanding the data flow, processing algorithms, and the sensitivity of the data being handled to ensure proper encryption, access controls, and resilience against potential attacks on the backend systems.* * *Are there industry regulations that need to be considered?* * *Compliance with Industry Regulations: Identify any industry-specific regulations or compliance standards applicable to the application. These might include GDPR for data privacy, PCI DSS for payment card data, or HIPAA for healthcare data. Ensuring adherence to these regulations is critical for avoiding legal ramifications and maintaining customer trust.* |
| **II. Define the technical scope** | List oftechnologies 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  The prioritization of technologies within the application hinges on their critical roles and functionalities. In this case:  1. \*\*Application Programming Interface (API):\*\* Given its fundamental role in facilitating interactions between different software systems, prioritizing API technology is essential. It enables seamless communication between diverse applications, fostering integration and functionality crucial for modern applications.  2. \*\*Public Key Infrastructure (PKI):\*\* PKI stands out due to its pivotal role in securing data transmissions and authentication processes. Prioritizing PKI ensures robust encryption, secure communication channels, and reliable identity verification, aligning with the heightened emphasis on data security and privacy in contemporary applications. |
| **III. Decompose**  **application** | [Sample data flow diagr](https://docs.google.com/presentation/d/1ol7y79popTFfNHM-90ES-H-i1Lpd0YNvPShxBlXozjg/template/preview?resourcekey=0-DZAkf7Vzh2PXsP-j3oXV-g)  [Am](https://docs.google.com/presentation/d/1ol7y79popTFfNHM-90ES-H-i1Lpd0YNvPShxBlXozjg/template/preview?resourcekey=0-DZAkf7Vzh2PXsP-j3oXV-g)  This data flow diagram illustrates a simplified process for a product search related to sneakers:  1. \*\*Product Search Process:\*\* Users initiate a search for sneakers for sale, which triggers a query.    2. \*\*Database:\*\* The search query accesses the database containing information about the current inventory of sneakers.    3. \*\*Listings of Current Inventory:\*\* The database responds to the query by providing listings of available sneakers that match the search criteria back to the user.  This basic representation demonstrates the flow of information from the user's search request to the database, which processes the request and returns the relevant inventory listings for sneakers available for sale. Typically, in a comprehensive data flow diagram, multiple processes, databases, and interactions between various components would be depicted, providing a more detailed view of the entire system. |
|  |  |
| **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?* * *Internal Threats: These threats originate from within the organization or system. Examples might include insider threats from disgruntled employees, accidental data leaks by employees, or inadequate access controls allowing unauthorized internal users to access sensitive information.* * *What are the external threats?* * *External Threats: These threats arise from sources external to the organization or system. External threats encompass a wide range of risks such as cyber attacks from hackers, malware infiltration, phishing attempts, denial-of-service attacks, and exploitation of system vulnerabilities by external actors aiming to compromise the application's data or functionality.* |
| **V. Vulnerability analysis** | List **2 vulnerabilities** in the PASTA worksheet that could be exploited.   * *Could there be things wrong with the codebase?* * *Codebase Vulnerabilities: Flaws within the application's code can be exploited. Examples might include insecure coding practices, lack of input validation leading to SQL injection or cross-site scripting (XSS) vulnerabilities, improper error handling, or insufficient encryption techniques.* * *Could there be weaknesses in the database?* * *Database Weaknesses: Vulnerabilities within the database might exist, such as poorly configured access controls, inadequate encryption of sensitive data, SQL injection vulnerabilities, or outdated database software with known security flaws.* * *Could there be flaws in the network?* * *Network Flaws: Weaknesses within the network infrastructure could pose vulnerabilities. These might include unpatched or outdated network equipment, misconfigured firewalls or routers, lack of encryption for data in transit, or inadequate network segmentation, potentially exposing sensitive information to unauthorized access.* |
| **VI. Attack modeling** | [Sample attack tree diagram](https://docs.google.com/presentation/d/1FmWLyHgmq9XQoVuMxOym2PHO8IuedCkan4moYnI-EJ0/template/preview?usp=sharing&resourcekey=0-zYPY7AhPJdcClXamlAfOag)  This simplified attack tree outlines potential threats to user data within the application:  1. \*\*User Data:\*\* Represents the main target or asset.    a. \*\*SQL Injection:\*\* Exploiting vulnerabilities in input fields to execute malicious SQL queries, potentially compromising the user data.    b. \*\*Session Hijacking:\*\* Exploiting weak session management to gain unauthorized access to user sessions and data.    c. \*\*Lack of Prepared Statements:\*\* Vulnerabilities arising from not using prepared statements, enabling attackers to manipulate SQL queries.    d. \*\*Weak Login Credentials:\*\* Potential risk due to weak passwords or insecure authentication methods, allowing unauthorized access to user accounts and their associated data.  Each branch of the attack tree signifies a potential avenue for attackers to exploit vulnerabilities within the application, leading to the compromise of user data. In reality, such attack trees are more extensive, encompassing various other attack vectors and potential exploits. |
| **VII. Risk analysis and impact** | List **4 security controls** that you’ve learned about that can reduce risk.  Certainly, here are four security controls that can effectively reduce risks:  1. \*\*Firewalls:\*\* Implementing firewalls acts as a barrier between a trusted internal network and untrusted external networks, filtering incoming and outgoing network traffic based on predefined security rules, thus preventing unauthorized access and potential threats.  2. \*\*Encryption:\*\* Utilizing encryption techniques for data both at rest and in transit ensures that even if attackers gain access to the data, it remains unreadable and unusable without the decryption key, significantly reducing the impact of a potential breach.  3. \*\*Multi-factor Authentication (MFA):\*\* Implementing MFA adds an additional layer of security by requiring users to provide multiple forms of verification to access systems or data. This significantly reduces the risk of unauthorized access, even if credentials are compromised.  4. \*\*Regular Software Patching and Updates:\*\* Consistently updating and patching software systems and applications helps address known vulnerabilities and weaknesses, reducing the likelihood of exploitation by attackers leveraging known security flaws. |