## PASTA worksheet // Zakhar Medunytsia

| **Stages** | **Sneaker company** |
| --- | --- |
| **I. Define business and security objectives** | ***Business Objectives:***   1. *Enable secure buying and selling of sneakers through a mobile platform.* 2. *Ensure seamless inventory browsing and product searching for end-users.* 3. *Provide transaction integrity and data confidentiality during purchases.*   ***Security Objectives:***   1. *Ensure protection of sensitive customer data (e.g., payment info, personal data).* 2. *Prevent unauthorized access to internal systems and databases.* 3. *Comply with industry regulations such as PCI-DSS for payment processing.* |
| **II. Define the technical scope** | **Technologies in Use:**   * **API (Application Programming Interface)** – used to connect front-end (app) to the back-end (server/database). * **Public Key Infrastructure (PKI)** – secures communication and authentication using digital certificates. * **SHA-256** – provides hashing for password storage. * **SQL** – manages the application's relational database.   **Priority Justification:**  We prioritize the **API and PKI** because they are integral to the core operations of the app: secure communications, identity verification, and data exchange. Given the sensitivity of financial and personal information, robust encryption and secure data handling must be ensured. SHA-256 and SQL are essential, but API misuse and certificate handling present more immediate attack surfaces. |
| **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** | ***External Threats:***   1. ***SQL Injection*** *– attackers could access or manipulate data via unfiltered user input.* 2. ***Session Hijacking*** *– attackers intercept valid session tokens to impersonate users.*   ***Internal Threats:***   1. ***Misconfigured access controls*** *– employees or internal systems may access sensitive data beyond their role.* 2. ***Codebase vulnerabilities*** *– insecure code deployment could open avenues for exploitation.* |
| **V. Vulnerability analysis** | ***Identified Vulnerabilities:***   1. ***Lack of Prepared SQL Statements*** *– as per the attack tree, this opens the system to SQL injection attacks.* 2. ***Weak Authentication Logic*** *– easily guessable passwords and session mismanagement make the system vulnerable to credential stuffing or hijacking.*   *Additional vulnerabilities may include:*   * *Inadequate logging or auditing.* * *Insecure data storage on client devices.* |
| **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** | **Recommended Security Controls:**   1. **Use of Prepared Statements and ORM Frameworks** – to prevent SQL injection attacks. 2. **Multi-Factor Authentication (MFA)** – strengthens login processes and protects against stolen credentials. 3. **TLS (HTTPS) Enforcement** – ensures data in transit is encrypted, mitigating risk of session hijacking. 4. **Regular Code Audits and Penetration Testing** – proactively identify and resolve vulnerabilities before exploitation. |