## PASTA worksheet

|  |  |
| --- | --- |
| **Stages** | **Sneaker company** |
| **I. Define business and security objectives** | * The app must process transactions securely. * Data privacy and responsible handling of user information are top priorities. * Proper payment handling is crucial to avoid legal issues. * *Users can create member profiles internally or by connecting external accounts.* * *The app must process financial transactions.* * *The app should be in compliance with PCI-DSS.* |
| **II. Define the technical scope** | List oftechnologies used by the application:   **API**: Facilitates exchange of data between users, partners, and employees. High risk for sensitive data exposure.   **Public Key Infrastructure (PKI)**: Provides encryption and authentication services.   **Advanced Encryption System (AES)**: Secures data at rest and in transit.   **SHA-256**: Used for hashing passwords or other sensitive data.   **SQL**: Stores data in relational databases, susceptible to SQL injection if not handled securely.  *I would prioritize evaluating the "Public key infrastructure (PKI)" technology. It involves the encryption of sensitive data and the exchange of keys, which are essential components for protecting user information and ensuring secure transactions in the mobile app. The use of symmetric and asymmetric encryption algorithms (AES and RSA) for data protection is a critical security consideration.* |
| **III. Decompose application** | A data flow diagram would include elements like:   * User → Application (via API) → Database (via SQL) * Third-party external accounts (OAuth or similar) * Payment gateway integration for financial transactions (API) * Encryption systems for securing sensitive data |
| **IV. Threat analysis** | List **2 types of threats** in the PASTA worksheet that are risks to the information being handled by the application.   * *Threat: Malware infection on the authentication system.* * *Threat: Social engineering attacks targeting employees involved in the authentication process.* * *Injection: Attackers could inject malicious code via forms or APIs, targeting SQL or other code vulnerabilities.* * *Session Hijacking: If an attacker intercepts or manipulates session tokens, they could impersonate a legitimate user, gaining unauthorized access.* |
| **V. Vulnerability analysis** | List **2 vulnerabilities** in the PASTA worksheet that could be exploited.   * *Vulnerability: Inadequate encryption of credit card information in the payment system.* * *Vulnerability: Insufficient input validation leading to potential SQL injection vulnerabilities.* * *Lack of Prepared Statements: If SQL queries are constructed dynamically, attackers can inject malicious code.* * *Broken API Token: If tokens are improperly validated or expire too late, attackers can reuse them to impersonate legitimate users.* |
| **VI. Attack modeling** | **Attack Tree Diagram**:   * Root: **Breach User Data**   + Node 1: **Exploiting SQL Injection**     - Leaf: **Bypassing input validation**   + Node 2: **Session Hijacking**     - Leaf: **Intercepting cookies or tokens**   + Node 3: **API Vulnerabilities**     - Leaf: **Exploiting weak API token handling** |
| **VII. Risk analysis and impact** | * Implement multi-factor authentication (MFA) to enhance user account security. * Conduct regular code reviews and penetration testing to identify and address vulnerabilities. * Implement data encryption for credit card information using strong encryption algorithms. * Employ web application firewalls (WAFs) to protect against SQL injection and other web-based threats. * **SHA-256**: Secure password hashing and data integrity verification. * **Incident Response Procedures**: Ensure that all security incidents are logged, monitored, and responded to effectively. * **Password Policy**: Enforce strong password policies to prevent weak passwords and brute-force attacks. * **Principle of Least Privilege**: Limit user and application privileges to only what is necessary, reducing potential attack surface. |