API Security Requirements

Instructions

As a team, evaluate the security requirements of an API of your choice and write a brief security requirements specification which mitigates against any risks associated with the API for enabling data sharing, scraping and connectivity between a program code written in Python and any of the following file formats/management systems (XML, JSON and SQL).

Remember to save your work to your GitHub repository and e-portfolio.

Learning Outcomes

- Identify and manage challenges, security issues and risks, limitations, and opportunities in data wrangling.
- Critically analyse data wrangling problems and determine appropriate methodologies, tools, and techniques (involving preparing, cleaning, exploring, creating, optimising and evaluating big data) to solve them.
- Systematically develop and implement the skills required to be effective member of a development team in a virtual professional environment, adopting real life perspectives on team roles and organisation.

Security Requirements Specification for a Fitness Tracker API

Overview

This document outlines the security requirements for an API enabling data exchange between a fitness tracker and a mobile phone. The API facilitates the transmission of health data, including metrics such as heart rate, step count, and sleep patterns. The data may also be synchronised with cloud services or third-party applications. To ensure security, the API must address risks such as unauthorised access, data breaches, and compliance violations.

Security Requirements

To protect the fitness tracker API from security threats, the following measures are required:

<u>Authentication and Authorisation</u>

Authentication and authorisation are essential to ensure that only authorised devices and users can access sensitive health data.

- Enforce mutual authentication during the pairing process, such as using secure tokens or QR codes.
- Implement OAuth 2.0 for secure integration with third-party applications.
- Apply role-based access control (RBAC) to manage permissions for data sharing and synchronisation.

<u>Data Transmission Security</u>

To prevent interception of sensitive data, secure transmission protocols must be employed.

- Use Bluetooth Secure Connections with AES-128 encryption for communication between the fitness tracker and mobile phone.
- Encrypt all API communications with HTTPS using TLS 1.2 or higher.
- For particularly sensitive data, apply application-layer encryption before transmission.

Input Validation and Sanitisation

All inputs to the API must be validated and sanitised to prevent injection attacks and other processing vulnerabilities.

- Validate inputs against predefined schemas to ensure they meet expected formats.
- Reject oversized or malformed payloads to avoid buffer overflows and denial-of-service (DoS) attacks.
- Sanitise inputs for device identifiers, user data, and firmware updates to prevent code injection.

Data Storage Security

Sensitive health data must be stored securely on both the fitness tracker and mobile phone.

- Encrypt health data at rest using AES-256 encryption.
- Store API keys and sensitive information securely using mechanisms such as the Android Keystore or iOS Keychain.
- Rotate encryption keys periodically and manage them using secure key management systems.

Rate Limiting and Abuse Prevention

To prevent abuse or overuse of the API, mechanisms to limit requests must be in place.

- Implement rate limiting to restrict the number of API requests per device or user.
- Detect and throttle suspicious activity patterns, such as repeated failed requests or excessive synchronisation attempts.

Logging and Monitoring

Effective logging and monitoring are necessary to detect unauthorised access and unusual behaviour.

- Log API activity, including device identifiers, user actions, and timestamps.
- Redact sensitive data, such as user identifiers and health metrics, in logs to protect privacy.
- Monitor logs for suspicious activity, such as bulk data transfers or repeated authentication failures.

Firmware and Software Updates

Firmware and software updates must be handled securely to prevent tampering or exploitation.

- Ensure all firmware updates are digitally signed and verified for authenticity.

- Use secure over-the-air (OTA) update mechanisms to protect against tampering.
- Perform compatibility checks to ensure updates do not disrupt the integration between the fitness tracker and mobile app.

Privacy and Compliance

The API must comply with data privacy regulations and prioritise user consent.

- Obtain explicit user consent before collecting or sharing health data.
- Provide users with controls to manage their data, including options to delete or export it.
- Anonymise or aggregate data for non-user-specific analysis to comply with GDPR and similar regulations.

Error Handling

Error responses must avoid exposing sensitive system details to unauthorised parties.

- Return generic error messages, such as "Invalid request," to prevent the disclosure of system information.
- Log detailed error information securely for debugging and troubleshooting purposes.

Format-Specific Security Measures

The API must ensure secure handling of data formats commonly used in communication and storage:

- JSON: Validate payloads against strict schemas and enforce size limits to prevent memory exhaustion or DoS attacks.
- XML: Disable external entity processing to prevent XML External Entity (XXE) attacks. Apply strict schema validation and size restrictions to ensure safety.
- SQL: Use parameterised queries or an ORM (e.g., SQLAlchemy) to prevent SQL injection. Limit database access permissions to authorised applications or processes.

High-Level Security Strategies

The following strategies should guide the overall implementation of the API:

1. Secure Communication Protocols: Employ Bluetooth Low Energy (BLE) and HTTPS with TLS for secure data transmission.
2. Regular Security Testing: Conduct penetration tests, code reviews, and vulnerability assessments to identify and mitigate security risks.
3. User Education: Provide users with guidance on secure pairing, privacy policies, and managing their health data.
4. Secure Tools and Libraries: Use cryptography libraries such as PyCryptodome for encryption and Bleak for secure Bluetooth communication.

Summary

This specification addresses the key security concerns for an API used to exchange data between a fitness tracker and a mobile phone. By implementing robust measures for authentication, encryption, input validation, and compliance, the API will ensure the confidentiality, integrity, and availability of sensitive health data. These practices align with industry standards, foster user trust, and create a secure environment for data-sharing and synchronisation.