Security Requirements Specification for API

Mahir Hashimov
12695120

Unit 10
Deciphering Big Data
University of Essex Online
23 June 2024

Contents

1. Introduction

The purpose of this study is to provide the security specifications for an API that would enable data exchange, scraping, and connection between a Python programme and several data management systems, particularly XML, JSON, and SQL. In order to provide strong security for ABC Electronics' data management procedures, the goal is to reduce the risks related to data breaches, illegal access, and data integrity problems.

2. Security Objectives

Ensuring the availability, confidentiality, and integrity of data transferred via the API are the main security goals. This entails guarding the API against typical online vulnerabilities and attacks, blocking unauthorised access, making sure that data is accessible only by authenticated and authorised users, and putting in place logging and monitoring to identify and handle security events.

3. Security Requirements

3.1 Authentication and Authorization

Authorization and authentication are essential to the API's security. Each client requesting access to the API will need to authenticate themselves using secure techniques like JWT (JSON Web Tokens) and OAuth 2.0. This guarantees that the API can only be used by authorised users. Furthermore, role-based access control (RBAC) will be implemented to guarantee that users may only access resources that they are authorised to use, as determined by clearly defined roles and the permissions that go along with them (Taipalus, 2024 and Kameswari et al., 2024).

3.2 Data Transmission Security

The API will encrypt data while it is in transit between the client and the server using Transport Layer Security (TLS) to safeguard data during transmission. Every API endpoint will have HTTPS support, guaranteeing that data is secured while being sent. Strong encryption techniques like AES-256 for data at rest and TLS 1.2 or higher for data in transit are among the encryption standards that will be employed. This will shield data from prying eyes and guarantee its confidentiality and security (Hossain et al., 2024 and Mirayala, 2024).

3.3 Data Integrity

To guarantee that the data transferred over the API is valid and unmodified, data integrity must be upheld. To stop injection attacks like SQL injection, XML External Entity Injection, and JSON injection, the API will strictly validate all inputs. This will entail verifying that the information complies with predetermined standards and formats. In order to confirm that the data has not been altered, checksums or hashes (such as SHA-256) will also be utilised to confirm the integrity of the data transferred via the API (Cuzzocrea et al., 2011 and Panda & Patra, 2015).

3.4 Data Storage Security

The API will make sure that sensitive data is secured using robust encryption techniques in order to keep it from being kept in databases. The implementation of database access controls will limit user access to data while upholding the concept of least privilege, which guarantees that users have the minimal amount of access required to carry out their duties. By doing this, illegal access will be prevented and data security and confidentiality will be maintained (Ma & Wang, 2024 and Hossain et al., 2024).

3.5 Input and Output Handling

It is essential to handle data input and output correctly to avoid security flaws. The API will ensure that special characters are correctly escaped or eliminated, sanitising all inputs to prevent injection attacks. By doing this, harmful data won't be handled. To ensure that the data provided to users is safe and secure, output encoding will also be used to avoid cross-site scripting (XSS) attacks when displaying data in user interfaces (Cuzzocrea et al., 2011 and Panda & Patra, 2015).

3.6 API Rate Limiting and Throttling

Rate limitation will be used to safeguard the API from misuse and denial-of-service (DoS) attacks. Based on use patterns, this will include placing restrictions on the maximum number of requests that may be made to the API in a given amount of time. To guarantee equitable use and avoid depletion of resources, throttling techniques will also be implemented. Under high load circumstances, these steps will assist in preserving the API's performance and availability (Mirayala, 2024 and Kameswari et al., 2024.

3.7 Logging and Monitoring

To identify and address security events, thorough tracking and monitoring are necessary. All requests and answers, including authentication attempts, data access, and error messages, will be meticulously logged by the API. To aid with activity monitoring and analysis, these logs will contain user IDs and timestamps. Using technologies like intrusion detection systems (IDS) and security information and event management (SIEM) systems, real-time monitoring will be set up to identify and address suspicious activity or possible breaches (Taipalus, 2024 and Ma & Wang, 2024).

3.8 Vulnerability Management

To find and fix any possible security flaws in the API and the underlying infrastructure, regular security audits and vulnerability assessments will be carried out. Patch management will lessen the chance of known vulnerabilities being exploited by ensuring that security updates and patches are quickly deployed to the API software and its dependencies (Panda & Patra, 2015 and Mirayala, 2024).

3.9 Incident Response

Manage security breaches with an efficient incident response strategy. A thorough incident response plan detailing the actions to be performed in the case of a security breach will be included with the API. In order to ensure that events are managed efficiently and that regular operations are resumed as soon as feasible, this will comprise protocols for containment, eradication, recovery, and communication and Cuzzocrea et al., 2011).

3.10 Compliance

The General Data Protection Regulation (GDPR) for data protection and the Payment Card Industry Data Security Standard (PCI DSS) for payment data security are only two examples of pertinent laws and standards that the API will abide with. This will entail taking the required steps to guarantee that data is managed in compliance with legal and regulatory obligations, safeguarding the company and its clients (Hossain et al., 2024).

4. Conclusion

By putting these security measures in place, the dangers connected with the API will be reduced and safe data exchange, scraping, and communication between a Python programme and XML, JSON, and SQL data management systems will be ensured. ABC Electronics will need to regularly examine and update these criteria in order to keep up with new security risks and preserve a dependable and safe data management environment.

References

- Cuzzocrea, A., Song, I.-Y., & Davis, K. C. (2011). Analytics over large-scale multidimensional data: the big data revolution! Proceedings of the ACM 14th International Workshop on Data Warehousing and OLAP, 101-104. DOI: 10.1145/2064676.2064695
- Hossain, M.A., Akter, S., Yanamandram, V., & Strong, C. (2024). Navigating the platform economy: Crafting a customer analytics capability instrument. Journal of Business Research, 170. DOI: 10.1016/j.jbusres.2023.114260
- Kameswari, P., Ramesh, V., Bhavikatti, I., & Gondesi, G. (2024). Analyzing the role of big data and its effects on the retail industry. Web Intelligence, 22(2): 1-19. DOI: 10.3233/WEB-230027
- Ma, X., & Wang, Z. (2024). Computer security technology in E-commerce platform business model construction. Heliyon, 10(7). DOI: 10.1016/j.heliyon.2024.e28571
- Mirayala, N.K. (2024). Evolving trends in open-source RDBMS: Performance, scalability and security insights. International Journal of Science and Technology, 12(2). DOI: 10.21275/SR24126224648
- Panda, M., & Patra, M. R. (2015). Data wrangling: Techniques and challenges of big data. International Journal of Applied Research, 1(10): 997-1001. DOI: 10.5441/002/edbt.2016.44
- Taipalus, T. (2024). Database management system performance comparisons: A systematic literature review. Journal of Systems and Software, 208. DOI: 10.1016/j.jss.2023.111872