Team 3: ReviewPro API

ReviewPro is a tool used by nearly all hotels in the hotel industry to analyse online guest reviews and other guest feedback to enhance guest satisfaction and online reputation. Their API allows developers to integrate ReviewPro's data and functionality into their own applications.

Security Requirements Specification:

- Authentication: The ReviewPro API uses API keys for authenticating requests. These keys should be securely stored, never exposed publicly, and rotated periodically to prevent unauthorized access.

- Data Encryption: All communication between the client and the API should be encrypted using HTTPS to protect the data during transit and to guard against eavesdropping and man-in-the-middle attacks.

- Use a web application firewall: Implementing a WAF to help detect and block common web-based threats.

- Rate Limiting: API endpoints are among the growing list of DDoS targets. Thus, it is important to implement rate limiting to prevent abuse and potential denial-of-service (DoS) attacks. For example, setting a threshold above which subsequent requests will be rejected (for example, 10,000 requests per day per account) can prevent denial-of-service attacks.

- Input Validation: Thorough validation of all user inputs and API parameters should be enforced to prevent injection attacks and other forms of malicious input.

- Injection flaws (including SQL injection, NoSQL injection, and command injection) involve data that is sent to an interpreter from an untrusted source via a command or query. Attackers can send malicious data to trick the interpreter into executing dangerous commands or allow the attacker to access data without the necessary authorization.

- Secure Data Handling: Since the API handles data in formats like JSON and XML, secure practices for data parsing and serialization should be implemented to prevent data tampering or injection attacks.

- Logging and Monitoring: All API requests and responses should be logged and regularly audited for any suspicious activities. Implement an anomaly detection system to identify potential security threats and trigger alerts.

- Inadequate logging, monitoring, and incident response integration can be used by attackers to stay in a system longer, gain a stronger grip, and steal or destroy more data. Given that it often takes more than 200 days to identify a persistent threat and that breaches are frequently found by a third party, robust API monitoring is crucial.

- Access Control: Implement strong access control measures to ensure that users can only access and modify data that they're authorized to. A common practice is the principle of least privilege. This foundational security principle holds that subjects (users, processes, programs, systems, devices) be granted only the minimum necessary access to complete a stated function. It should be applied equally to APIs.

- Regular Security Audits and Updates: Conduct regular security audits to identify any potential vulnerabilities in the API. Patch and update any identified security risks promptly.

- Identity: The API should implement secure authentication mechanisms, such as OAuth or API keys, to verify user identity. Access to the API should be restricted to authorized individuals or systems with appropriate access controls.

- Data Validation: The API must validate and sanitize user input to prevent security vulnerabilities like SQL injection and cross-site scripting attacks.

- Secure Communication: All communication between the Python program and the API should be conducted over HTTPS to encrypt data transmission and protect against eavesdropping.

- Data Privacy: The API should handle personal and sensitive data securely. Data should be encrypted both at rest and in transit. Compliance with relevant data protection regulations, such as GDPR, should be ensured.

Adherence to these security requirements will help ensure that the ReviewPro API is securely handling data sharing, scraping, and connectivity between Python code and various file formats and management systems.