Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 10, No. 2, 2024

API Common Security Threats and Security Protection Strategies

Chenchen Zhao *

Boston University, Boston, Massachusetts, United States
*** Corresponding author Email:** zhao900805@gmail.com

Abstract: This study analyzes the core role of APIs in modern digital ecology and the security threats they face, such as information leakage and overstepping access, and explores their security risks for technologies such as RESTful and GraphQL. It proposes to use OAuth/JWT authentication mechanism to strengthen access control, adopt HTTPS/TLS to secure data transmission, and combine with API gateway to defend against DDoS attacks. It also emphasizes the importance of fine-grained privilege management and log auditing. The study provides strategic guidance for improving API security protection and looks forward to the trend of intelligent protection.

Keywords: API Security; Security Protection Strategy; OAuth/JWT Authentication; HTTPS/TLS Protocols.

1. Introduction

API (Application Program Interface), as a key bridge for data and service interaction, plays an increasingly important role in the field of Web, mobile applications and back-end system integration, and has become an indispensable core component for the construction of digital ecosystem in the information age. With the in-depth development of the trend of "Internet Plus", APIs are not only integrated into the products themselves, but also evolve into a key carrier for enterprise value realization and business interconnection. According to statistics, between 2018 and 2023, the market size and growth rate of China's API services market are both showing an upward trend (see Figure 1). However, the rapid progress of Internet technology has also brought increasingly severe cybersecurity challenges. APIs have become high-frequency targets for cyberattacks due to their intermediary nature of data transmission, and their security is crucial to system stability and data protection. According to the API Security Landscape Report 2024 published by Imperva, 71% of Internet traffic involves API calls, making APIs a direct conduit for accessing sensitive data. Fastly's survey data reveals that 95% of organizations have experienced API security issues in the past year, while Marsh McLennan's research further reveals that API related security incidents cost global organizations up to $75 billion annually. Against this backdrop, securing APIs has become an important issue in maintaining the stability of the network ecosystem.

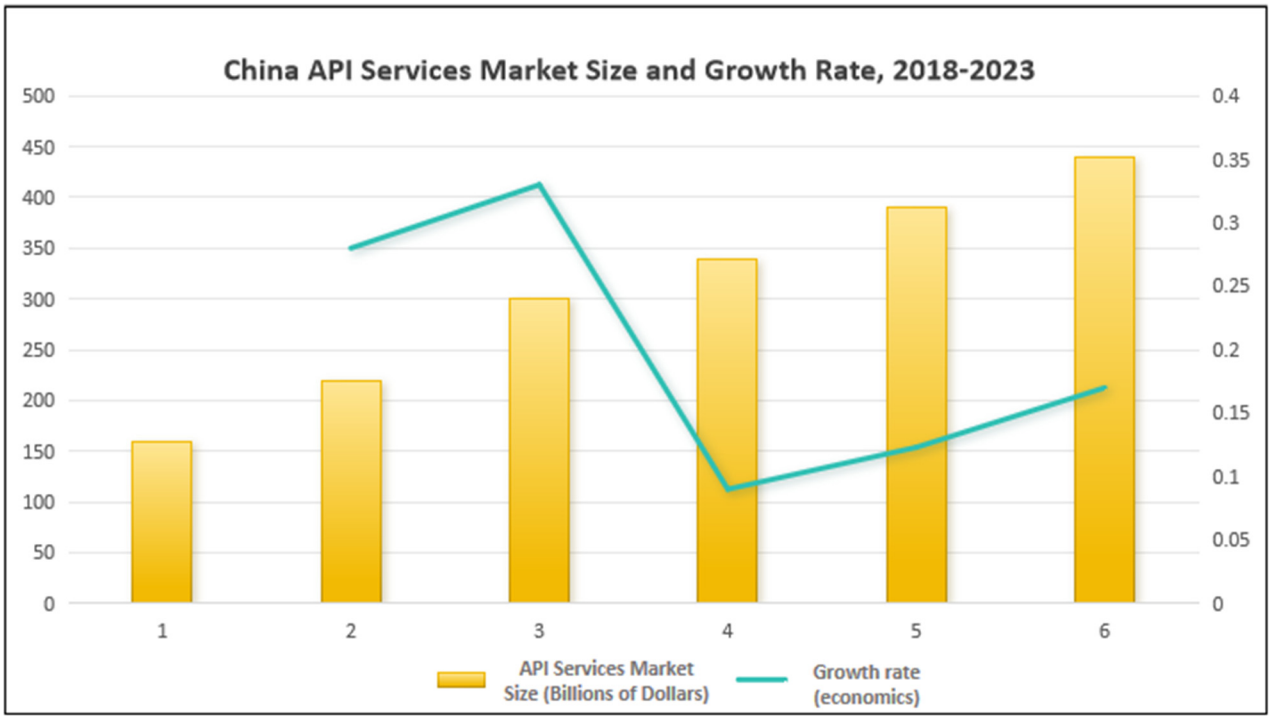

Figure 1. China API Services Market Size and Growth Rate, 2018-2023
(Data source: https://blog.csdn.net/gyzlsc/article/details/134238884)

2. API Basic Concepts and Technology Overview

2.1. API Definition and Role

API, or Application Programming Interface, occupies a central position in software development as a bridge for interaction between different software applications. It enables developers to call functions or services of other applications without knowing the details of the underlying implementation, thus promoting modular design and significantly improving code reusability and maintainability. In modern Web services, APIs serve as a key medium for data interaction between the front-end and the back-end, realizing the dynamic data update of the user interface. For example, the GitHub API allows developers to access and control multiple resources on the GitHub platform through HTTP requests, greatly expanding the scope of its functional applications. At the same time, APIs also promote the development of cross-platform applications, enabling the same set of back-end services to support a variety of front-end devices such as cell phones, tablets and computers. To ensure the uniformity, ease of use and scalability of the interface, the design of APIs needs to follow specifications and principles such as RESTful, which in turn realizes the flow of data between different systems and promotes the prosperity and development of the software ecosystem.

2.2. API Common Technologies and Risks

In the field of API technology, RESTful, GraphQL, gRPC, and SOAP are the four mainstream technologies. RESTful API, which has appeared since 2000, is based on the HTTP protocol, supports XML and JSON, and provides a unified interface service for WEB, IOS, and Android, etc., which is widely used in client-server interaction and has become the most widely used API technology. However, in microservice architecture, such as SpringBoot's Actuator module, if not properly configured, it is easy to lead to security vulnerabilities (see Figure 2). graphQL is an open-source data query language that provides efficient and flexible API development, but the implementation is complex, and the introspection feature and multitasking request function are security risks. gRPC, as a high-performance RPC framework based on the HTTP/2 and ProtoBuf, supports multiple languages, and is suitable for mobile and industrial interconnections. gRPC, as a high-performance RPC framework based on HTTP/2 and ProtoBuf, supports multiple languages, and is suitable for mobile and industrial interconnections. gRPC is a strictly defined information exchange protocol, which has been widely used due to its features, but is gradually decreasing in newer systems due to its slowness and inefficiency. improperly configured SOAP interfaces can also lead to leakage of the WSDL file, which can trigger security risks. Therefore, developers need to choose appropriate API technologies based on comprehensive consideration of application scenarios, technical characteristics and security risks, and strengthen security measures to ensure the security and stability of APIs to cope with potential security challenges [1].

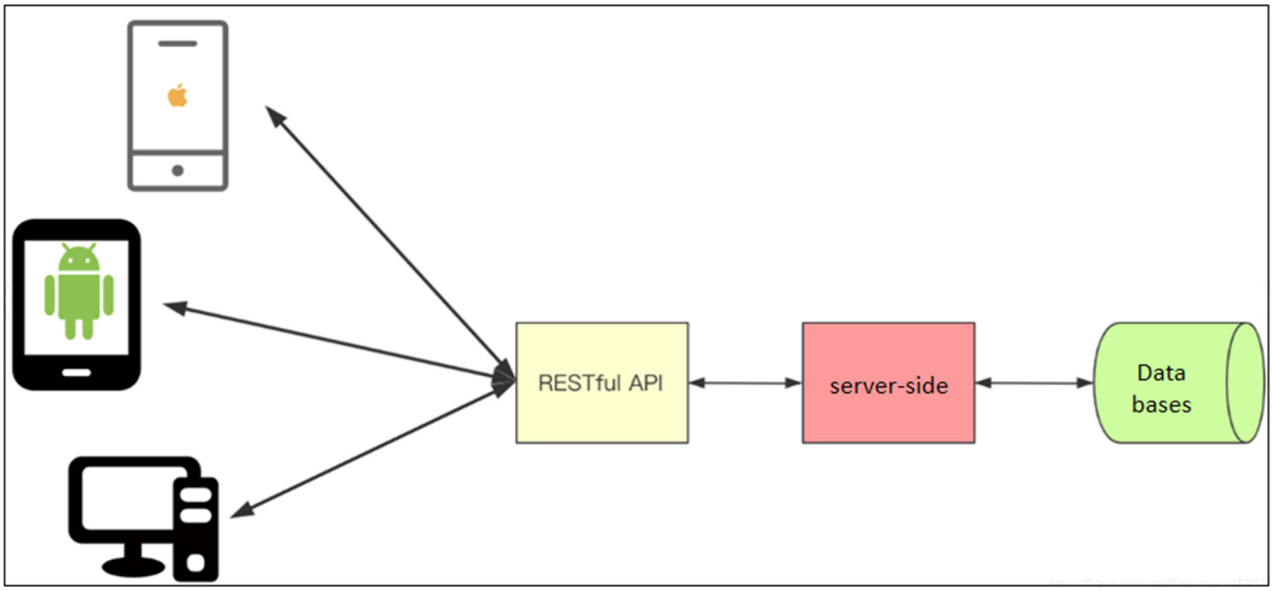

Figure 2. Example of restful api interface specification

3. API Common Security Threat Analysis

3.1. Information Leakage

In the security analysis of API interface, information leakage is a problem that cannot be ignored. Under normal circumstances, users should only be able to access APIs corresponding to the scope of their privileges, and attackers often try to detect and access APIs outside the scope of the user's privileges, or APIs that the user does not need to access in the first place [2]. In practice, API access control for legitimate users is often neglected, resulting in large or comprehensive abnormal exposure of APIs (Figure 3). Especially in microservice development, the extensive use of API gateway facilitates interface and application management, but also raises many internal security issues. Inadequate access control settings between internal logics can easily lead to unauthorized access and information leakage. For example, in Springboot-based development, the security protection of API visualization schemes such as Swagger and Actuator is often overlooked, resulting in the full exposure of the interface and the leakage of sensitive information such as

monitoring data and configuration files. In recent years, Druql, Jenkins, WordPress and Spring boot and other mainstream development frameworks have occurred in the API interface leakage of serious security issues, highlighting the importance of strengthening API security protection.

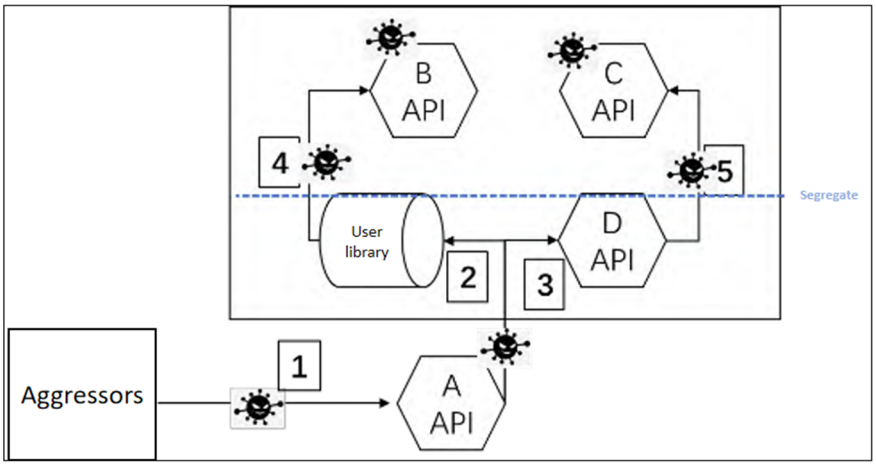

Figure 3. Information Disclosure Vulnerability

3.2. Unauthorized Access

In the field of API security, ultra vires access is a key issue, which is rooted in the diversity of API authentication logic and incomplete implementation [3]. APIs usually use HTTP request headers with an independent token field for authentication, but due to the strong independence of the APIs and the designers' poor considerations in the implementation of different API functions, it is possible to utilize the differences in the implementation of APIs to gain ultra vires access. However, due to the strong independence of APIs and the poor consideration of designers in the implementation of different API functions, it is possible to utilize the differences in the implementation of APIs to carry out unauthorized access. There are two types of unauthorized access: horizontal unauthorized access and vertical unauthorized access. Horizontal overstepping occurs after the user has passed API authentication. Since the designer has not fully considered the isolation of privileges between users, the attacker can utilize the identity of user A to construct a malicious request to obtain the business data of user B, C, etc. Vertical transgression occurs when the attacker utilizes the identity of ordinary user A to construct malicious requests to access the management API. If the management API does not take appropriate security restriction measures, it may expose the management functions to ordinary user A, or even indirectly attack the management API through certain application APIs as a proxy, resulting in a more serious security threat.

3.3. Lateral Movement and Injection Attacks

Within the operation mechanism of API gateway, the data interaction between internal applications and components is based on the principle of trust, and the APIs are connected to each other through database, message queue, function call and API proxy forwarding, etc., and these connections often lack additional security restrictions and protection measures. Once an attacker obtains access to a function or module, he or she can use the internal data exchange function to construct a malicious request and insert it into the data flow, thus modifying, stealing or controlling the data of other APIs and realizing the so-called lateral movement attack (see Figure 4). Given that most of the current API functions are realized in the form of Web services, common vulnerabilities in the Web domain, such as SQL injection, XSS attacks, etc., may also appear in the API. If the API fails to validate and filter the input data effectively, the attacker has the opportunity to construct malicious code to obtain, modify or delete sensitive information in the database.

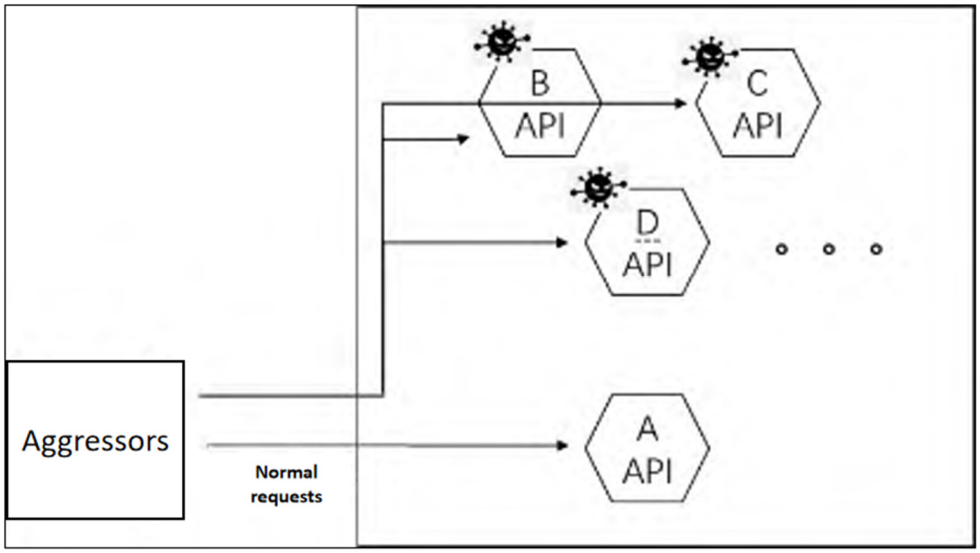

Figure 4. Lateral movement

4. API Security Protection Strategy and Practice

4.1. Access Control and Authentication

The core of API security protection lies in access control and authentication, in which OAuth and JWT play a key role as the mainstream authentication mechanism [4] (see Figure 5). OAuth distributes access tokens through the authorization server to achieve a fine-grained control of access to resources, and effectively prevents unauthorized access, such as Amazon e-commerce platforms have adopted this mechanism to protect the security of user data. JWT is compact and self-contained to achieve stateless authentication. JWT, with its compact and self-contained characteristics, realizes stateless authentication, especially adapted to distributed systems, but its signature algorithm and key management need to be strictly controlled to prevent forgery and tampering. Fine-grained privilege management is also critical to ensure that only authorized users have access to sensitive resources by assigning privileges to API endpoints, as successfully practiced in Google Cloud Platform. The combination of API gateway and policy engine can realize dynamic privilege control and enhance the flexibility and adaptability of security protection. Therefore, the reasonable application of OAuth, JWT and other authentication mechanisms, and the implementation of fine-grained rights management is an effective strategy to enhance API security and protect data and system integrity.

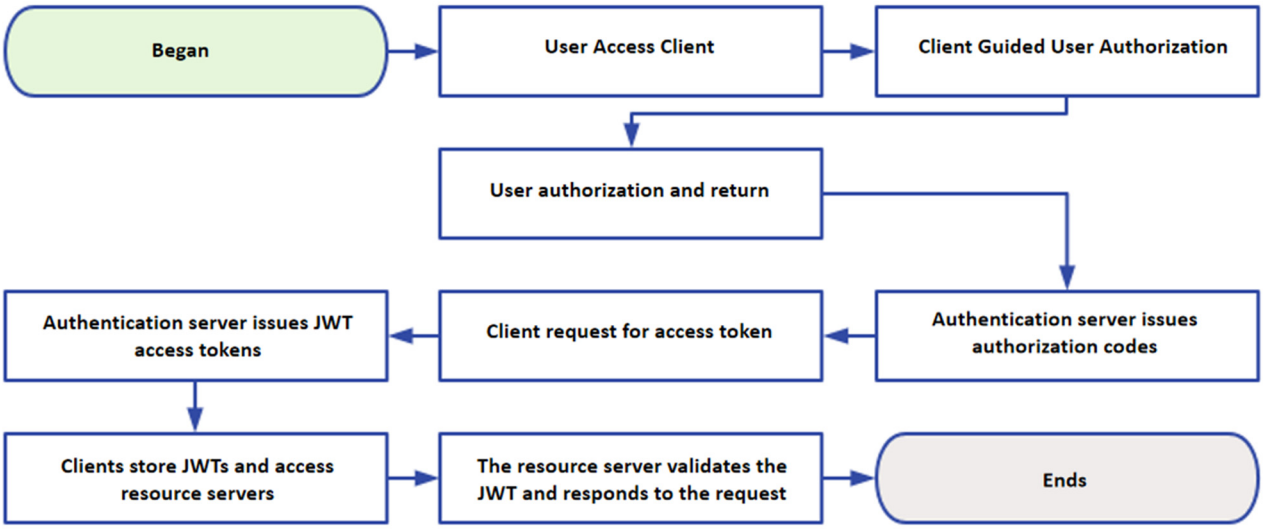

Figure 5. Two-Factor Authentication Flow with OAuth and JWT Combination

4.2. Data Encryption and Transmission Security

HTTPS and TLS protocols, as a security standard for Internet communication, are widely used in API transmission to ensure data confidentiality, integrity and reliability. HTTPS establishes an encrypted channel between the client and the server through the SSL/TLS protocol, effectively preventing the data from being stolen or tampered with during the transmission process, e.g., the PayPal online payment platform adopts this mechanism to guarantee the secure transmission of user for example, PayPal online payment platform adopts this mechanism to guarantee the safe transmission of user payment information. For the storage of sensitive data, the use of encryption technologies such as AES, RSA, etc. is crucial to ensure the security of data at rest, and medical and healthcare systems such as Mayo Clinic further safeguard the security of patient privacy by applying the TLS protocol during data transmission. The effective combination of HTTPS, TLS protocol and encryption technology not only significantly improves the security of the APIs but also provides an effective solution for the storage and transmission of sensitive data. provides a solid protective barrier for sensitive data in storage and transmission.

4.3. Security Protection, Hardening and Backup Strategy for API Gateway

API gateway bears the heavy responsibility of traffic management, flow limiting and DDoS defense in microservice architecture. It can intelligently route requests, achieve load balancing, and prevent service paralysis through rate limiting [5]. At the same time, the API gateway recognizes and filters malicious traffic in DDoS attacks to protect back-end service security. At the security hardening level, log auditing and anomaly detection are the core, with comprehensive logging providing the basis for security event traceability, while the anomaly detection system monitors API calls in real time to identify potential threats. Through its API gateway service, combined with log auditing and anomaly detection mechanisms, Amazon successfully fended off multiple DDoS attacks to safeguard business continuity and user data security, highlighting the practical utility of API gateways and reinforcement measures. Utilizing the simple and rapid deployment of API, a backup mechanism can be established. The API topology is constructed according to the internal invocation of the system API (see Figure 6), and the replacement and reinforcement program for each node after the API is controlled is formulated. When the API is attacked or isolated, it can be quickly deployed to replace the controlled API and reinforce other APIs with the same function to prevent lateral penetration by attackers.

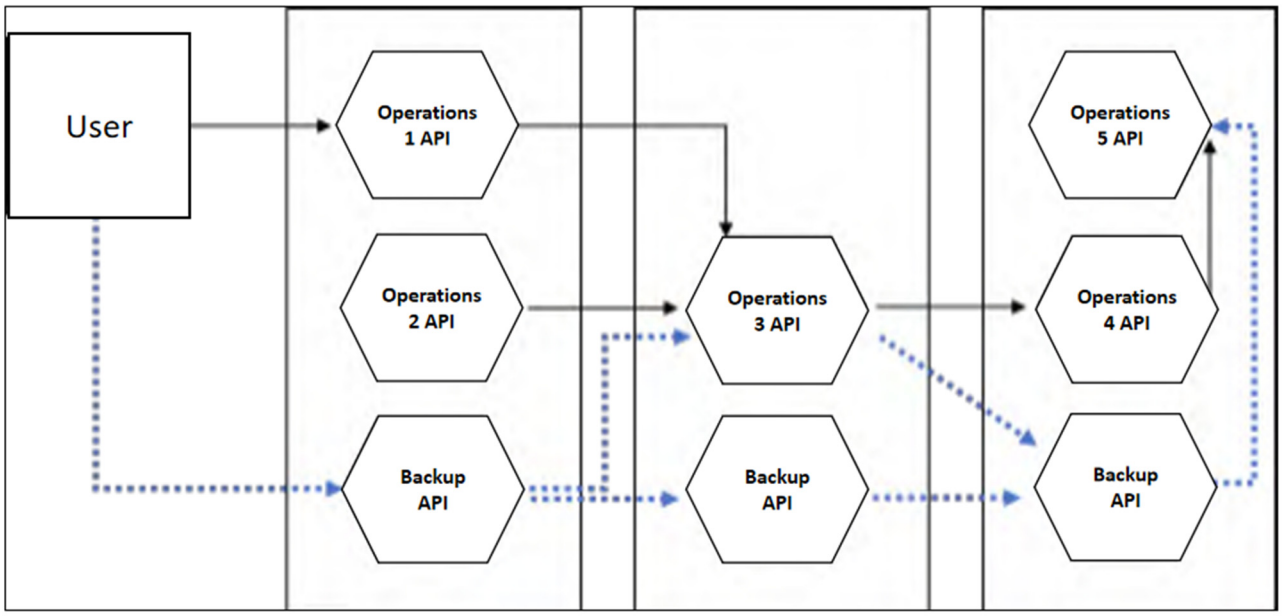

Figure 6. API backup topology

5. Conclusion

AP This paper comprehensively examines the key position of APIs in modern digital architectures and the security threats they face, such as information leakage and ultra-rights access, and deeply analyzes the security challenges of mainstream technologies such as RESTful and GraphQL in practical applications. Through the introduction of OAuth, JWT authentication mechanism and fine-grained rights management, the access control of API is effectively enhanced and data leakage is prevented. Meanwhile, the application of HTTPS and TLS protocols ensures the security of data transmission. In the future, with the development of AI and ML technology, API security protection will tend to be intelligent, improving adaptive and predictive capabilities. The optimization of API gateway in traffic management, defense against DDoS and other functions will further enhance the comprehensiveness of API security protection. Therefore, continuous innovation in API security protection is crucial to maintaining the stability and security of the digital world.

References

[1] Hu Hongyu. Application of cloud WAF in hospital critical web service and API protection[J]. Network Security Technology and Application, 2024, (09):137-139.

[2] Huang Jian. Implementation and research of API application security in software development environment[J]. Post and Telecommunications Design Technology, 2024, (08):39-43.

[3] YAN Jiwei, HUANG Jinhao, YANG Hengqin,et al. Anomaly-sensitive framework API lifecycle model construction [J/OL]. Journal of Computing,1-21[2024-09-18].http://kns. cnki. net/ kcms/ detail/11.1826. TP.20240626.0922.002.html.

[4] B. Luo, C. Guo,G. W. Shen,et al. A ransomware early detection method based on API latent semantics[J]. Electronic Journal, 2024, 52(04):1288-1295.

[5] Wang Lei. Design and realization of enterprise application integration system based on API gateway[D]. Guangzhou University, 2024.