International Journal of Engineering & Scientific Research
Vol.11 Issue 04, April 2023
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Secure APIs - Best Practices and Measures

Sunil Kumar Rangineni

Abstract

In this article, we will discuss API Security measures and best practices in securing the API's or webservices.

About Author:A Seasoned Cyber security professional with 15 plus years of experience in the Information Security domain, with profound knowledge, focusing on solving organizations' pain points in the Cyber Threat landscape.

Proven track record in securing critical infrastructure, such as global financial markets and pharmaceutical Industries, against evolving cyber risks. I have a deep knowledge of security standards and regulations, such as the Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPAA), and have experience in developing and implementing security policies.

As a cybersecurity enthusiast, my main goal is to raise awareness about cybersecurity within organizations. My aim is to contribute to the improvement of overall security awareness and assist in enhancing security measures.

Introduction:

An API (Application Programming Interface) acts as an intermediary between two distinct software applications, enabling seamless communication and data exchange. By providing a standardized interface, APIs offer developers the ability to access specific functionalities or data from another software application or service without the need to understand or modify the underlying code. This results in more efficient development processes, improved interoperability between applications, and enhanced overall functionality.APIs represent a potent resource for developers as they offer a uniform approach to accessing data and functionality from various software applications and services, resulting in the creation of more efficient and effective software solutions. This not only streamlines the development process but also enhances the overall performance and scalability of the resulting applications.

As the usage of APIs continues to rise, it is imperative that they are appropriately secured. Many industry-wide threats are the result of excessive or sensitive data being leaked through APIs. To mitigate this risk, it is essential to implement a shift left approach, which involves securing APIs from the development stage to maintenance. By integrating security measures at the earliest stage of the development process, developers can reduce the risk of potential vulnerabilities being introduced into the API. Additionally, regular maintenance and updates can ensure that any new vulnerabilities are identified and promptly addressed. Overall, implementing a shift left approach is critical to ensuring the security and protection of data being transmitted through APIs.

Type of API's: An Overview

APIs are categorized in multiple types, depends on the functionality and access. Some of the most commonly used are:

Rest APIs: Representational State Transfer (REST)APIs are the most commonly used type of API, enabling communication between diverse software systems using standard HTTP requests and responses.

SOAP APIs: Simple Object Access Protocol (SOAP) APIsare a form of web service that utilize XML-based messaging protocols for transmitting data between applications. These APIs rely on the Remote Procedure Call (RPC) protocol, which enables applications to request services from each other on a network.

GraphQL APIs: GraphQL APIs provide developers with a distinct approach to REST APIs by enabling them to define the precise information they require, instead of retrieving a determined dataset. These APIs utilize a query language and a private endpoint to retrieve information from a server.

OpenAPI/Swagger APIs: OpenAPI/Swagger APIs are used to define and document RESTful APIs, allowing developers to understand the structure and functionality of the API before using it.

What is API Security?

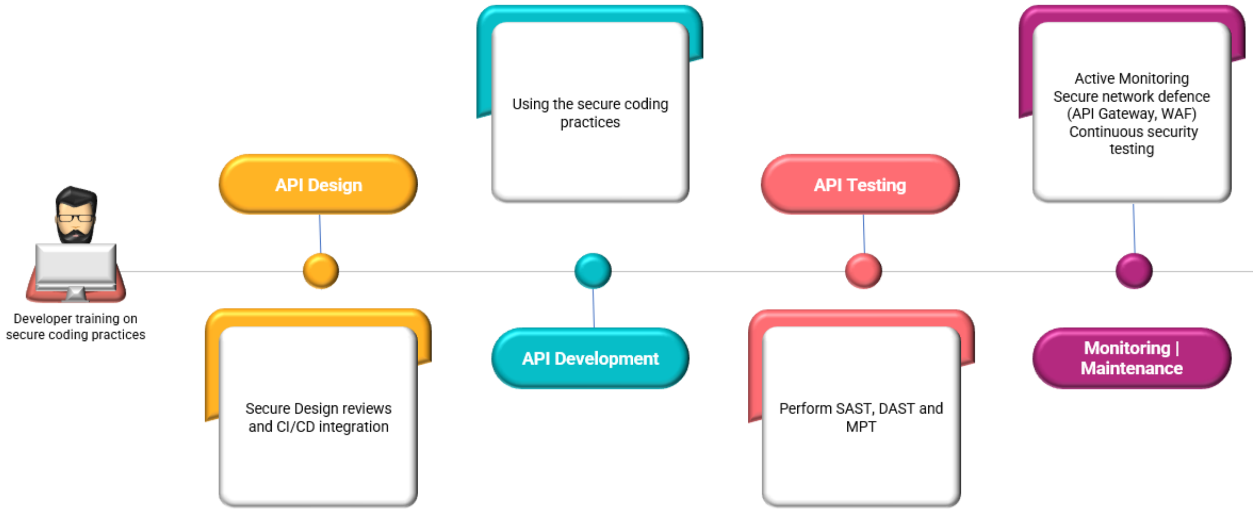
API security encompasses a range of measures and best practices aimed at safeguarding APIs and the data transmitted through them against unauthorized access, exploitation, and misuse. Due to their vulnerability to attacks, such as injection attacks and unauthorized access, APIs are particularly susceptible to data breaches and sensitive data exposure. Effective API security protocols aim to prevent such incidents and ensure that APIs remain secure throughout their lifecycle, from development to maintenance.

By implementing and adhering to best practices in API security, organizations can ensure that the APIs they use and develop remain secure and that sensitive data transmitted through them is protected from unauthorized access and exploitation. This, in turn, helps to safeguard the overall security and integrity of the organization's digital systems and protects against potentially devastating data breaches.

Standards Used: There are multiple standards and frameworks used to secure APIs. Some of the widely used are: Open Web Application Security project (OWASP) API Security Top 10, National Institute of standardsand technology (NIST) API Security, Open-standard Authorization protocol (OAUTH), Open ID connect. **Reference:**https://brightsec.com/blog/api-security/

API Security Components:

Note: The following components have been captured with a focus on security and no other types of testing or processes have been included.


Developer training on secure coding practices
API Design
Secure Design reviews and CI/CD integration
Using the secure coding practices
API Development
API Testing
Perform SAST, DAST and MPT
Active Monitoring
Secure network defence (API Gateway, WAF)
Continuous security testing
Monitoring | Maintenance

API Security Common Challenges:

Following are some of the common challenges faced when it comes to API Security:

- Ensuring only authorized users and applications can access the API is a major challenge in API security, requiring proper **authentication and authorization** mechanisms.

- API security must address the **risk of sensitive data exposure**, including the use of encryption to protect against loss or leakage.
- **3rd Party Library Vulnerabilities**: The use of third-party libraries and components in APIs introduces vulnerabilities that attackers may exploit. Organizations must update and patch these components regularly to prevent exploitation.
- API security must defend against **Denial of Service (DoS) attacks**, which involve overwhelming the API with a flood of requests to disrupt its normal operation. DoS protection mechanisms are essential to prevent such attacks.
- **Injection attacks** are a serious threat to API security, as attackers can insert malicious code into API requests to exploit vulnerabilities. Organizations must implement input validation and sanitization techniques to prevent these attacks.

Why do we need API security?

APIs are the foundation of modern digital ecosystems, enabling seamless communication and data exchange between various applications, services, and devices. However, APIs are also a common target for threat actors due to the sensitive information they carry. Here are some of the reasons why API security is essential.

- Protect Sensitive data
- Mitigate Cybersecurity Risks
- Ensure Compliance
- Maintain Business Continuity
- Maintain Trust

In the recent times we have observed, several recent API security related attacks, some of them are:

SolarWinds Supply Chain Attack: During December 2020, it was uncovered that the SolarWinds Orion API had been breached, resulting in the insertion of malevolent code into updates that were disseminated to customers. This enabled cybercriminals to infiltrate the sensitive data and systems of numerous government agencies and businesses.

Reference:https://www.solarwinds.com/securityadvisory

Facebook API Bug: Hackers exploited a Facebook API bug in April 2021, which led to the unauthorized access of personal data belonging to more than 500 million users. This vulnerability enabled attackers to scrape users' birth dates, email addresses, phone numbers, and other confidential information.

Reference:https://www.cnn.com/2021/04/03/tech/facebook-data-breach-533-million-intl-hnk/index.html

T-Mobile Data Breach:T-Mobile announced a data breach in August 2021, affecting more than 50 million customers. The cybercriminals exploited an API vulnerability to gain entry to customers' personal data, including social security numbers, birth dates, names, and addresses.

Reference:https://www.t-mobile.com/news/business/t-mobile-confirms-unauthorized-access-to-some-customer-information

considering the constantly evolving threat landscape, securing APIs is of utmost importance to ensure the reliability, privacy, and security of modern digital environments.

Best practices to be considered for API Security

API security is critical to ensure that the application programming interfaces (APIs) of an organization remain protected from cyber threats. Here are some best practices to be considered for API security:

- Use strong authentication: use of strong authentication mechanisms (Eg: OAuth 2.0, OpenID Connect, or API keys) can help prevent unauthorized access to APIs
- Implement Proper role-based access controls
- Encrypt the data: Data at rest or transit to be encrypted
- Use HTTPS: Use secure channel for all API communications
- Implement rate limiting, Prevents API Abuse by limiting number of requests
- Follow Secure coding practices: Prevents most of the injection attacks
- Real time monitoring: Real-time monitoring of API traffic can help identify anomalies that could be indicative of a security breach.
- Security testing Program: to identify vulnerabilities and to ensure that APIs are secure.
- Incident response plan: To respond quickly to any security incidents that occur
- Vulnerability Management: Regular cadence of applying security patches and updates

API Security (Testing Standards)

Multiple security frameworks are in use, some of the common frameworks are:

Center for Internet Security (CIS) Controls framework which provides security controls that organizations can implement to improve their overall security posture.

National Institute of Standards and Technology (NIST) Cybersecurity Framework: a framework that outlines standards, guidelines, and best practices to help organizations manage and mitigate cybersecurity risks. This framework includes a specific emphasis on product security and offers a versatile approach to managing cybersecurity risks.

Open Web Application Security Project (OWASP) API: Framework for developing secure APIs. I wanted to emphasize more related to this framework As I see this as a base for APIsecurity. The OWASP Top 10serves as a guide for developers and organizations to prioritize and address security risks in their APIs.

Below listed is the OWASP API TOP 10 standards list:

The OWASP Top 10 API Security Risks is a list of the top ten most critical security risks for APIs, as identified by the Open Web Application Security Project (OWASP). The current version is OWASP Top 10 API Security Risks 2019.

- **Broken Object Level Authorization**: When an API fails to authenticate or authorize requests based on the required level of access to perform a particular action, this results in the occurrence of this issue.
- **Broken Authentication and Session Management**: The term pertains to security concerns that emerge from the erroneous implementation of authentication and session management. These concerns may include weak passwords, session fixation, and session hijacking, among others.
- **Excessive Data Exposure**: This occurs when an API returns too much information in response to a request, including sensitive or confidential data.
- **Lack of Resources & Rate Limiting**: The term encompasses problems such as inadequate rate limiting or insufficient allocation of resources. Such problems can lead to excessive resource consumption or denial-of-service attacks.
- **Broken Function Level Authorization**: This term pertains to problems concerning the authorization and access controls of specific functions or actions within an API.
- **Mass Assignment**: This happens when an API permits users to submit data in large quantities without appropriate validation or filtering. This can lead to the unauthorized manipulation of data and access to sensitive information.
- **Security Misconfiguration**: The term refers to security setting misconfigurations that can create vulnerabilities.
- **Injection**: This term pertains to problems related to the injection of malicious code or SQL queries into an API's request or response.

- **Improper Assets Management**: This issue arises when an API fails to manage its assets correctly, including security credentials and keys, or when it permits unauthorized access to these assets.
- **Insufficient Logging and Monitoring**: The term relates to problems regarding inadequate monitoring and logging of API activity, including events that may signal a breach or an attack.

Reference: https://owasp.org/www-project-api-security/

Conclusion:

Therefore, it is crucial to secure APIs against these threats. By implementing proper API security measures, organizations can protect their critical data and systems, mitigate cybersecurity risks, ensure compliance with regulatory requirements, and maintain business continuity.