RAPIFUZZ™

Securing Your APIs

Demystifying API Security

Part 3

Contents

1. Executive Overview

Organizations are moving from traditional, monolithic web applications to modern applications that utilize a microservices architecture. This results in smaller distinct units of functionality and often results in an explosion of web APIs to interact with those microservices. Application Programming Interface (APIs) are the gateways to these applications and carry sensitive data and if these APIs are compromised or hacked, they could lead to major data breaches. A study by RapidAPI[1] indicates some statistics on the increased use and consumption of APIs within applications.

- Seventy percent of developers indicated the increase in the usage of APIs in 2023.
- Sixty Three percent of developers reported a greater usage of APIs than they did in 2021.
- Ninety percent of developers indicated their intent or plan to test APIs
- Approximately 40% of the largest corporations (with over 10,000 employees) possess in excess of 250 internal APIs.

Nearly 64% of the smallest businesses (with 1-50 employees) utilize up to 10 internal APIs.

The report also states that the most common API type for developers was internal APIs, and they are consuming a lot of third-party APIs to help drive digital transformation. Nearly 25% to 30% of the respondents were focused on he acceptance, integration, and functional testing of APIs while less than 7% did API security testing.

Organizations like Netflix have been embracing an API-first approach since 2012, and they've shared how APIs offer them the flexibility to support more than 1,000 different device types. API-first[2] is defining and designing APIs and underlying schema before developing dependent APIs, applications, or integrations. It often involves constructing microservices and resharing their functionality internally. APIs bring reusability for internal components and help organizations decouple front-end and back-end development to work on more digital platforms simultaneously. This makes for shorter and more agile development cycles.

2. What Are APIs?

An API, or application programming interface, defines the protocols for communication among software components. Wikipedia[3] defines it as "*An application programming interface (API) is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build such a connection or interface is called an API specification.*

A computer system that meets this standard is said to implement or expose an API. The term API may refer either to the specification or to the implementation."

APIs define the rules that programmers must follow to be able to interact with a programming language, a software library, or any other software tool. These Web APIs are a set of rules for interacting with the web server, with the most common use case being data retrieval. APIs provide mechanisms for an end user of software to access and manipulate data stored by the API provider. The user makes a “request” to the software webserver, which then accesses the software’s database (with the customer’s data), and returns it to the requester in a “response”. This same request/response cycle is used when you access web pages in your browser. The major difference between an “API request” and a “webpage request” is that a website returns HTML, CSS, and JavaScript, which work together with your browser to render a web page. In contrast, a Web API responds with data in a raw format, which is not intended to be rendered by a browser into a user experience. JSON and XML are the most common formats used for this raw data, and they are both flexible text formats for storing data.

Present-day software programs are modular and use APIs to communicate with each other which can be used locally or remotely; these programs could either run on the same computer or machines that are separated by multiple time zones. In either case, the programs need a well-defined standard for exchanging data. In order for each to send data in a format that the other can understand, the same predefined protocol must be followed by both.

3. The Rising Significance of API Security

APIs play a very important role in the digital transformation journey and strategies of organizations. Securing these APIs is a priority and a top challenge as they are normally accompanied by new types of security vulnerabilities with an expanded attack surface. APIs are a rapidly growing attack surface that isn't widely understood and can be overlooked by developers and application security managers resulting in vulnerabilities slipping away and being exposed by hackers.

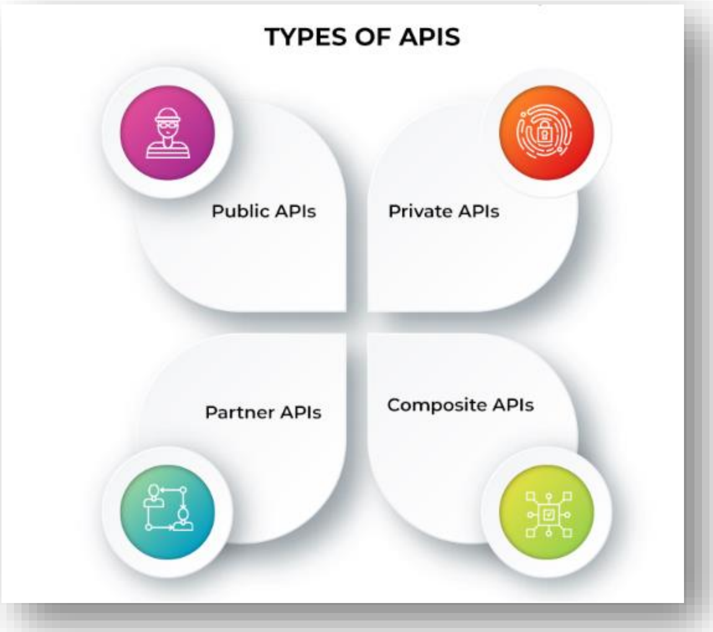
API security is a major focus and the lack of it poses a major challenge for organizations. Organizations need to be geared to address API security and develop capabilities to be able to automatically discover APIs and conduct API-specific testing than depending on traditional web application security technologies.

According to OWASP API Security Project[4], "From banks, retail, and transportation to IoT, autonomous vehicles and smart cities, APIs are a critical part of modern mobile, SaaS and web applications and can be found in customer-facing, partner-facing, and internal applications. By nature, APIs expose application logic and sensitive data such as Personally Identifiable Information (PII) and because of this have increasingly become a target for attackers. Without secure APIs, rapid innovation would be impossible."

4. Formats of APIs

APIs are not restricted to any particular format. Based on their accessibility and usage, APIs can be of three forms:

- **Private APIs**: APIs can be private. APIs that are only used internally can be categorized as Private APIs.
- **Semi-Public APIs**: APIs can be semi-public. In other words, although they are used in a public context (such as sending data over the Internet), their internal details are restricted to trusted entities.
- **Public APIs**: APIs can be public. Many applications and services publish their APIs so that external entities can communicate with them.



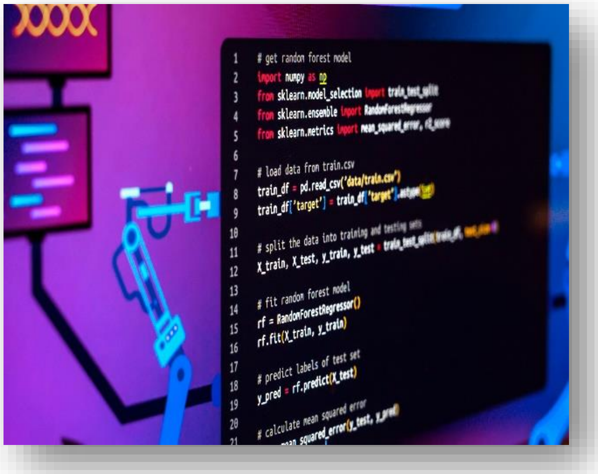
Today, there are three categories of API protocols or architectures: REST, RPC, and SOAP. These might be dubbed "formats," each with unique characteristics and tradeoffs and employed for different purposes. Another widely used is GraphQL, which is not an API itself, but rather a query language for APIs. It is a technology that allows you to interact with APIs in a more flexible and efficient way, compared to traditional REST APIs.

5. Types of APIs

Based on architecture and protocol the APIs can be classified as follows:

Here are some common types of APIs:

1. **RESTful APIs**: (Representational State Transfer): These are a type of web APIs that adhere to the principles of REST. They use standard HTTP methods (GET, POST, PUT, DELETE) and typically return data in JSON or XML format.



```
# get random forest model
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# load data from train.csv
train_df = pd.read_csv('data/train.csv')
train_df['target'] = train_df['target'].astype(int)

# split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(train_df, [illegible]

# fit random forest model
rf = RandomForestRegressor()
rf.fit(X_train, y_train)

# predict labels of test set
y_pred = rf.predict(X_test)

# calculate mean squared error
    mean_squared_error(y_test, y_pred)
```

2. **SOAP APIs:** (Simple Object Access Protocol): SOAP is a protocol for exchanging structured information in the implementation of web services. It uses XML as the message format and can be transported over various protocols, including HTTP, SMTP, and more.

3. **GraphQL APIs**: As mentioned earlier, GraphQL is a query language for APIs that allows clients to request only the data they need, making it highly flexible and efficient.

4. **JSON-RPC and XML-RPC APIs**: These are remote procedure call (RPC) protocols that use JSON or XML to encode requests and responses. They are used for invoking methods or functions on a remote server.

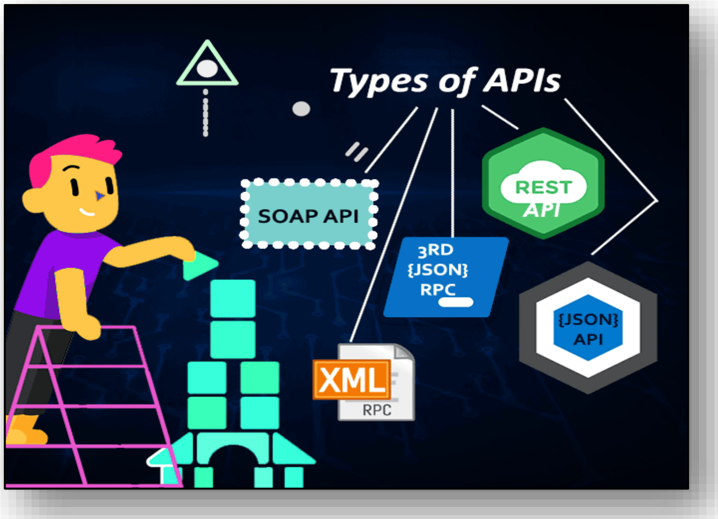
5. **GraphQL.** It is a query language for APIs that was developed in response to REST APIs, with the idea that you could execute precise syntax that

retrieves only what is needed, lightening the payload and simplifying the process significantly.

6. **gRPC**: RPC stands for Remote Procedure Call and refers to something that can execute a function housed elsewhere but in a different context. A user on one side will select a remote procedure to execute, serialize the necessary parameters, and then append any additional information to the message.

7. **WebSockets**. It is a communications protocol that provides dynamic communication over a single TCP connection.

There is no "best" API type as each has its own quirks and will be suited to particular applications. The choice of API type depends on the specific requirements and constraints of your project

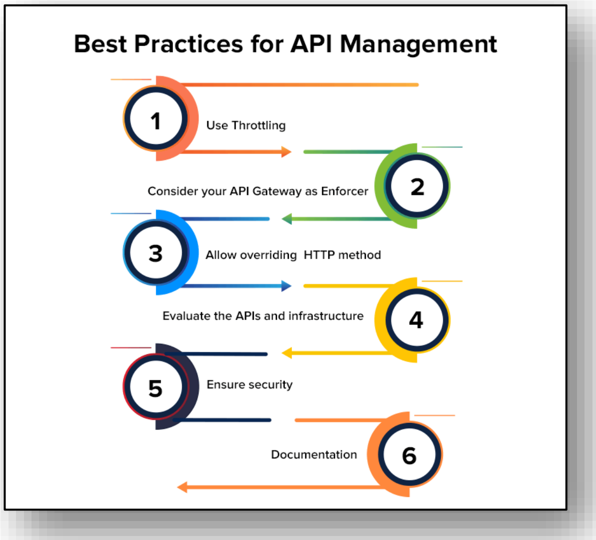


6. Unlocking the Merits of API Security Testing

6.1 Promotes Faster Innovation

The main reason that APIs matter so much in modern markets is that they allow faster innovation. APIs reduce barriers to change allowing more people to contribute to an organization's success. They offer two-fold benefits: the company can create better products while standing out from the competition. APIs also make monetization easier and save time for quick deployments. From a technical standpoint, APIs allow the capabilities of one computer program to be used by another.

6.2 Backbone of Most Web Applications

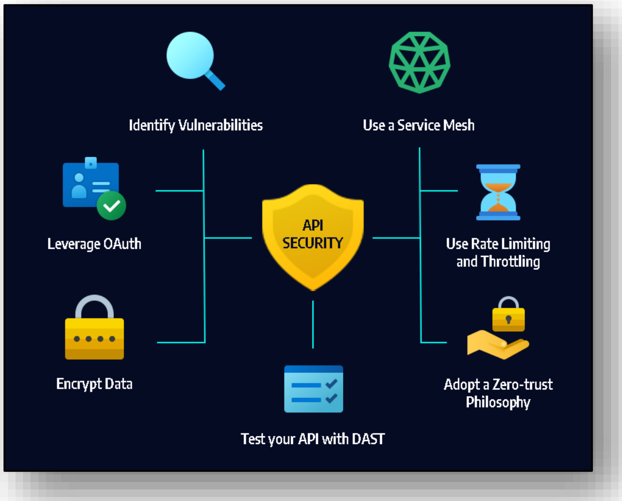

Best Practices for API Management
1
Use Throttling
2
Consider your API Gateway as Enforcer
3
Allow overriding HTTP method
4
Evaluate the APIs and infrastructure
5
Ensure security
6
Documentation

APIs are becoming the backbone of most web applications and the number of APIs has been exponentially increasing. As organizations use APIs to connect their services to other services and to transfer data, it is becoming important to ensure that security testing of APIs is done exhaustively. While creating APIs, it is important to ensure that they best practices are being followed and proper documentation is being maintained. API security is not just about their usage and deployment but also about testing them for any security vulnerability that could compromise the API. API security testing is slow, manual, costly, and requires a person with adequate knowledge of APIs.

6.3 Compromised APIs Result in Compromised Data

If the APIs are compromised then it would not just compromise the applications and organizations consuming them but also leak valuable data. For example, if an API is impacted by a Distributed Denial of Service (DDoS) attack, it would make the API and its associated service unavailable, leading to loss of revenue and a serious impact on the image of the application user. It is important to conduct security testing to ensure that rate limiting and throttling are implemented to safeguard APIs from such attacks.

API compromise can also result in data being stolen by hackers, competitors, or aggregators. APIs often deal with sensitive data, including user information, financial data, and business-critical information. Security vulnerabilities in APIs can lead to data breaches, which can result in severe financial and reputational damage. API security testing helps identify and mitigate these vulnerabilities before they can be exploited.



6.4 Promotes Authorized Access

Both legitimate users and potential attackers can access APIs. Security testing helps ensure that only legitimate users and systems can access the API, preventing unauthorized access and excessive data exposure. To ensure that legitimate users get access, it is important to implement proper authentication and authorization mechanisms. Security testing helps verify that these mechanisms are in place and functioning correctly to prevent unauthorized users from gaining access to sensitive data or excessive data or functionality.

6.5 Verifies Transmitted Data

APIs transmit data over the internet or other networks. It is important to do security testing to verify that data is transmitted securely, usually through the use of encryption (e.g., HTTPS), and to protect the data from eavesdropping or tampering during transit. API security testing can detect vulnerabilities that could allow attackers to inject malicious code or commands into API requests, potentially compromising the database or system.

6.6 Promotes Risk Mitigation

Business logic vulnerabilities can pose significant risks to an application's security and data integrity. Attackers often exploit these flaws to gain unauthorized access, manipulate data, or perform actions that violate security policies. Business logic evaluation provides insights into potential attack vectors that attackers might use to exploit the API. By testing the business logic, one can proactively identify and mitigate these risks before they can be exploited by malicious actors, reducing the potential impact of security incidents.

6.7 Reduces Legal and Financial Risks

Many industries have specific security compliance requirements (e.g., GDPR, HIPAA, PCI DSS). API security testing helps ensure that APIs comply with these regulations, reducing legal and financial risks. It is important to fix the API security bug at the initial stage rather than at a later stage, as the cost to fix the bug or the vulnerability increases. API security has now become vitally important for businesses today.

7. Conclusion

In the past year itself, major breaches have been reported where APIs have been compromised leading to major financial and reputation loss. All these, matched with new vulnerabilities being reported, make API security testing a must for every organization that exposes their APIs to external vendors. Because one only controls their own APIs, however, API security centers secures the APIs one exposes either directly or indirectly. API security is less focused on the APIs you consume than the ones that are provided b y other parties. It's also important to note that API security as a practice overlaps various teams and systems.

In modern software development, all solutions revolve around applications, which in turn revolve around APIs. APIs are everywhere, from your banking application, insurance, aviation, automobiles, gaming, entertainment, Oil, Power and Gas, IoT, critical infrastructure to smart cities, etc. All these applications expose their APIs to third-party integrators. The increase in the attack surface which in turn makes one's APIs vulnerable. A compromise in one solution could result in attackers finding ways to compromise other solutions based on a compromise. It is very important for organizations to have a strong process in place wherein they test every API which they consume before they are exposed to third party vendors.

8. References

- https://rapidapi.com/blog/state-of-apis-growth-and-more-growth-on-tap-for-2023
- https://swagger.io/resources/articles/adopting-an-api-first-approach
- https://en.wikipedia.org/wiki/API
- https://owasp.org/www-project-api-security/