DEMYSTIFYING API SECURITY

EXECUTIVE OVERVIEW

Organizations have substantially evolved due to the recent trends in software development, offering a dynamic and agile approach resulting in a shift from traditional monolithic web applications to modern applications, a loosely- coupled modular component known as microservices. This approach involves building application infrastructure with individual services resulting in smaller distinct units of functionality and often results in an explosion of web APIs to interact with those microservices. APIs are the gateways to these applications and carry sensitive data. If these APIs are compromised or hacked, they could lead to some major data breaches.

> IMVISION, Enterprise API Security Survey 2021, shows that more than 71% of large enterprises use more than 50 APIs whereas 80% of these enterprises publish APIs for either partners or clients.[1]

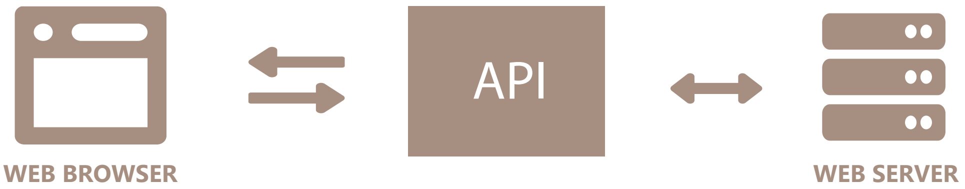
API developments are in general accompanied by new types of security vulnerabilities with an expanded attack surface. APIs play a primary role in the digital transformation journey and strategies of organizations. Securing these APIs is a priority and a top challenge. APIs are a rapidly growing attack surface that isn’t widely understood and might be overlooked by developers and application security managers. It may result in vulnerabilities slipping away and being exposed by hackers.

In response to the ever-shifting digital markets, business needs to adapt rather quickly, since here competitors can change a whole industry with nothing but a new app. To stay competitive, it’s really important to support the rapid development and deployment of new services, API security too, along with being a major focus poses some serious challenges 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 rather than depending on traditional web application security technologies.

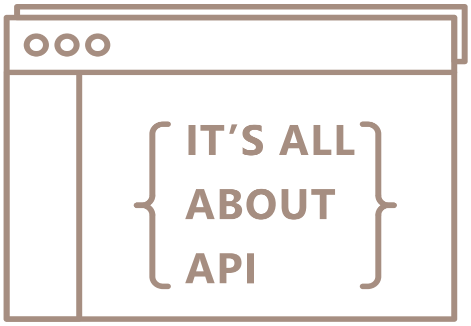
WHAT ARE APIs ?

An API, or application programming interface, defines the protocols for communication among software components. Wikipedia[2] 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.”



To interact with any programming language, software library, or any other software tool, programmers follow the set of rules defined by an API. These Web APIs acts as an intermediary layer interacting with the webserver. With data retrieval being the most common use case providing various mechanisms for an end-user to access and manipulate the data an API provider stored. The user can make a "request" to the software web server, which then accesses the software's database (with the customer's data),and returns it to the requester in a "response". This same cycle is in use when accessing web pages. However, 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 webpage. Whereas, in the case of Web APIs, they respond with data in a raw format like JSON or XML, the most common yet flexible text formats for storing data. The browser, however, does not intend to render this into a user experience.

According to OWASP API Security Project[3]- Top 10 2019, "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."



Present-day software programs are modular and use APIs to communicate with each other. Be it locally or remotely. It leads to a concern that the programs need a well-defined standard for exchanging data. They could be running from anywhere, either on the same computer or on machines, separated by different time zones. In either case, it's noted that each of them needs to send data in a format that the other can understand.

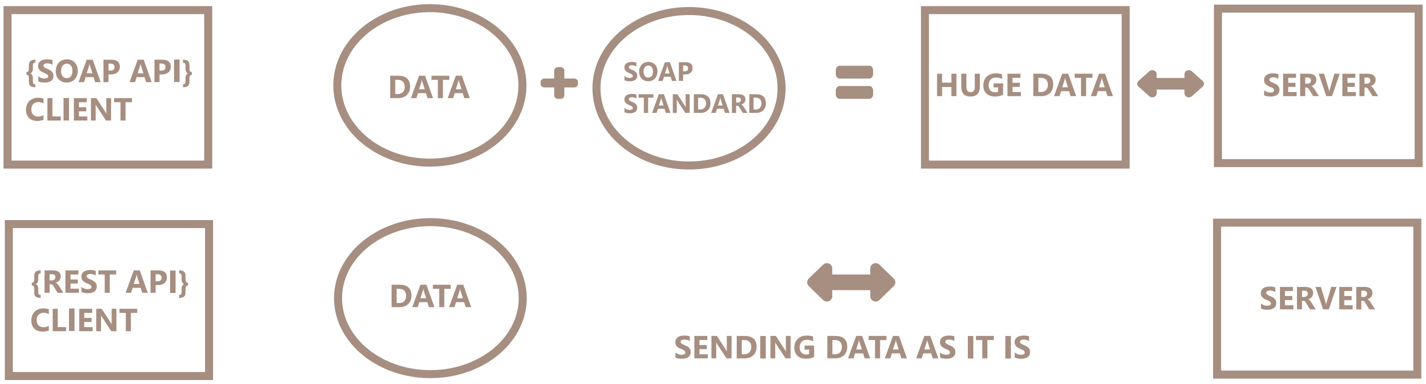
APIs can be private or semi-public and or used internally. For context, we can send data over the internet, where the APIs are in use publically. However, their internal details are restricted to trusted entities only. APIs are not bound to any particular format. However, REST and SOAP formats are the ones most commonly used today.

TYPES OF API

Based on architecture and protocols, the API’s can be classified as follows:

{ SOAP API }

Simple Object Access Protocol, also known as SOAP API is a protocol that is built with XML, which enables the users to send and receive data through SMTP and HTTP. With SOAP APIs, it becomes easier to share information between applications or any software component which is written in a different language or may or may not be are running in a different environment



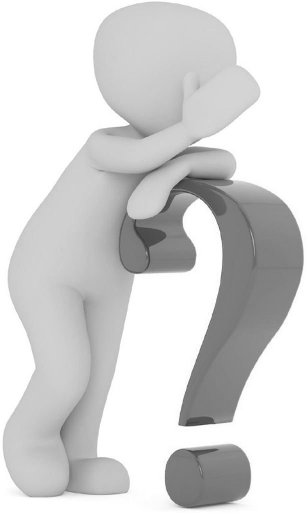
{ REST API }

Representational State Transfer also known as REST can be defined as a set of architectural principles defining a set of constraints that are used for creating web services. To be a REST API(RESTful API), web APIs must adhere to the REST architectural constraints. REST API intends to increase the desirable characteristics of inter-software communication such as performance, scalability, etc. followed by six underlying principles just to achieve this:

- ✓ Client-server architecture: It handles requests through HTTP. The architecture is composed of clients, servers, and resources
- ✓ Statelessness Information about the state of a session is held with the client since the client cant store content between requests
- ✓ Cacheability: The need to have some client-server interactions can be eliminated by caching.
- ✓ Layered system: We can mediate Client-server interactions by having additional layers. These layers can also offer features like load balancing, shared caches, or security.
- ✓ Code on demand: Servers can transfer executable code which can extend the functionality of a client
- ✓ Uniform interface: A core constraint to the design of RESTful APIs

WHY IS API IMPORTANT ?

Today's market focuses more on the modern approach of allowing dynamic innovation. API provides data this gives rise to the fact that there is no limit as to how anyone can go on about using that approach. With this, more people can contribute to an organization's success and the company can create better products while standing out from the competition.



APIs also make monetization easier, save time for quick deployments since it allows us to use the capabilities of one computer program by another. APIs are forming the backbone of most web applications giving rise to the number of APIs exponentially. As organizations use APIs to connect their services to other services and transfer data, it is crucial to ensure that security testing of APIs is done exhaustively. If the APIs get compromised, then it will compromise not just the applications and the organizations consuming them but also leak valuable data.

A vulnerability found in the BrewDog mobile app exposing users' PII courtesy of hard-coded bearer tokens.[4]

For example, if an API gets 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 significant impact on the image of the Application user.

API compromise can result in data getting stolen by hackers, competitors, or aggregators.These types of possible attacks made securing APIs difficult.
API security is not just about their usage and deployment but also about testing them for any security vulnerability which could compromise the API. API security testing is slow, manual, and costly, along with the fact that it requires a person with adequate knowledge of APIs.

It is crucial to fix the API bug at an initial stage rather than later, as the cost to fix the bug or the vulnerability increases. Today, API security has become vital for businesses. In modern software development, all solutions revolve around applications, which in turn revolve around APIs. Major applications expose their APIs to third-party integrators increasing the attack surface, which in turn makes the API vulnerable. A compromise in one single solution could result in attackers finding ways to compromise others. Giving us more than enough reason to find a strong process wherein an organization can test every API consumed by them before they get exposed to any third (3rd) party vendors.

Cited References
1. Imvison Enterprise API Security Servey. Retrieved from https://www.imvision.ai/2021-api-security-survey/
2. API's for What are API's. Retrieved from https://en.wikipedia.org/wiki/API
3. API Security Project- Top10, 2019. Retrieved from https://en.wikipedia.org/wiki/API
4. Issue 155: Vulnerability in BrewDog mobile app, October 13, 2021. Retrieved from https://apisecurity.io/

RAPIFUZZ

OCTOBER, 2021

TOSHI SAXENA (SECURITY EVANGELIST), AVNI TYAGI (DIGITAL MARKETING SPECIALIST)