

# API Terminology Handbook

The Ultimate Guide  
to Rest API Terms  
and Glossary



# Contents

- API
- API Call
- API Economy
- API Endpoint
- API Integration
- API Gateway
- API Lifecycle
- API Request
- API Keys
- API Layer
- API Portal
- API Security
- Apigee
- APIsec
- Application
- Framework
- Burp Suite
- CI/CD
- CRUD
- Cache
- Client
- DDoS
- Resource
- Request
- Response
- Response Code
- Payload
- Pagination
- Method
- Query Parameters
- Authentication
- Rate Limiting
- API Documentation
- Logic Flaw

# Contents

- JSON
- Microservices
- Monetization
- OWASP
- ZAP
- Parameters
- Penetration Testing
- Production Environment
- REST
- Red Teams
- SDK
- SDLC
- SOAP
- SQL Injection
- Webhook
- Over-Permissioned Container

END

# API

Application Programming Interface is what API stands for. API is a set of definitions and protocols that allow technology products and services to communicate via the internet.

## API Call

The API call is simply the process of sending a request to your API after setting up the right endpoints. Upon receiving your information, it is processed, and you receive feedback.

By entering your login and password into a website and hitting 'enter', you made an API call.

# API Economy

The API economy is a term to describe the **exchange of value** between a **user** and an **organization**.

It enables businesses to **leverage APIs** from other providers such as **Google** to power their own apps, allowing an **ecosystem** that makes it possible for users to get value from a platform **without** having to **build the APIs** from scratch.

For Example: Uber uses API calls to connect with Google Maps.

# API Endpoint

An endpoint is the **end of a communication channel**. When APIs interact with other systems, **each touchpoint of interaction** is considered an endpoint.

For example, it could be a server, a service, or a database where a resource lives.

# API Integration

In simple terms, API integration **connects** two or more applications to **exchange data** between them and connect to the **outside world**.

# API Gateway

An API gateway is an **API management tool** that serves as an **intermediary** between the client and a set of different backend services.

API gateways act as **gatekeepers** and **proxies** that moderate all your API calls, aggregate the data you need, and return the correct result.

Gateways are used to handle common tasks such as **API identification**, **rate limiting**, and **usage metrics**.

# API Lifecycle

The API lifecycle is an approach to API management and development that aims at providing **a holistic view of how to manage APIs** across its different life stages, from creation to retirement.

The API lifecycle is often divided into three stages, the **creation stage**, the **control stage**, and the **consumption stage**.

# API Request

APIs are everywhere and are part of every aspect of the web. An API request happens when a developer **adds an endpoint** to a URL and **uses that endpoint** to call the server or the database.

# API Keys

An API key is a **unique identifier** that enables other software to **authenticate** a user, developer, or API calling software to an API to ensure that this person or software is **who it says it is**.

API keys **authenticate the API** instead of a user and offer a certain degree of security to API calls.

# API Layer

An API layer is a **proxy** that joins together all your service offerings using a **graphic UI** to provide greater **user interactivity**. API layers are **language-agnostic** ways of interacting with apps and help describe the services and data types used to exchange information.

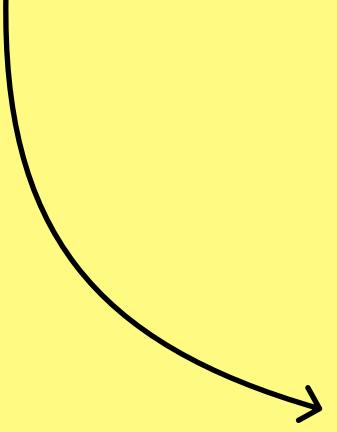
# API Portal

An API portal is a **bridge** between the **API provider** and the **API consumer**.

API portals serve to make APIs public and **offer content** to **educate developers** about them, their use, and how to make the most of them.

An API portal provides **information** about the **APIs** at every stage of the API lifecycle.

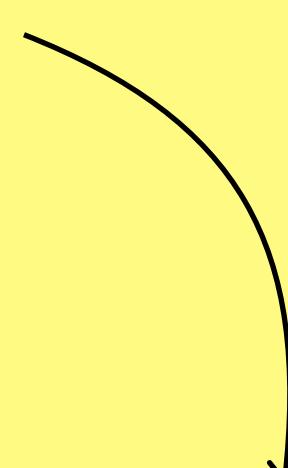
# API Security



API security is an umbrella term that defines a **set of practices** that aim to prevent malicious attacks, misuse, and exploit APIs.

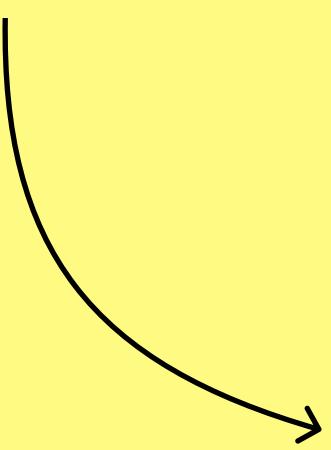


API security includes basic **authentication** and **authorization**, **tokens**, **multi-factor authentication**, and other advanced security measures.

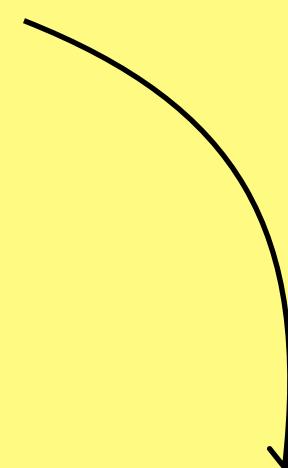
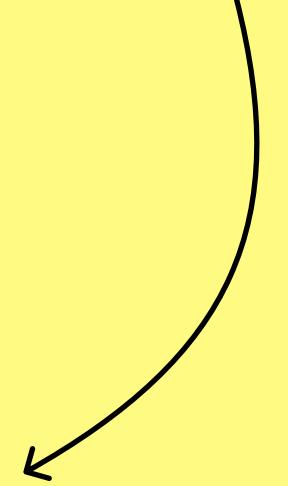


The ubiquitous nature of APIs makes them one of the **favorite targets** for hackers.

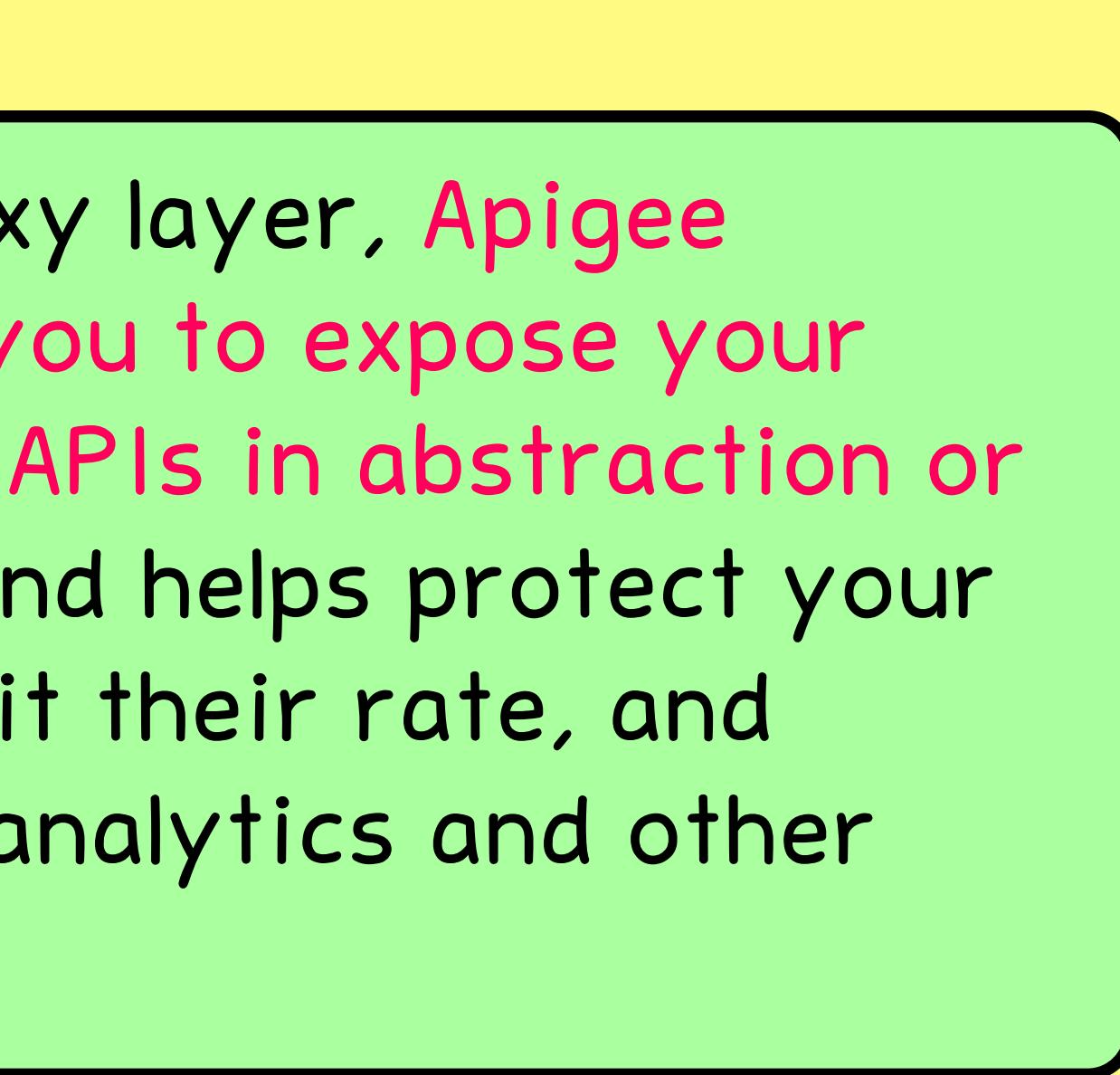
# Apigee



Apigee is an API **gateway** management tool offered by Google to **exchange data** across cloud services and applications.

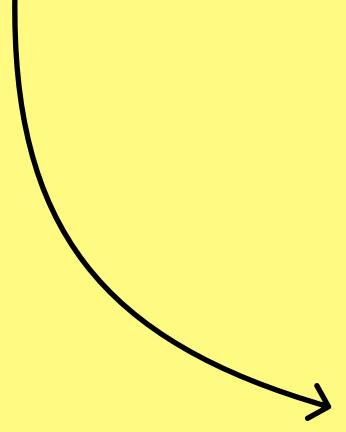


As a proxy layer, Apigee enables you to expose your backend APIs in abstraction or **facade** and helps protect your APIs, limit their rate, and provide analytics and other services.

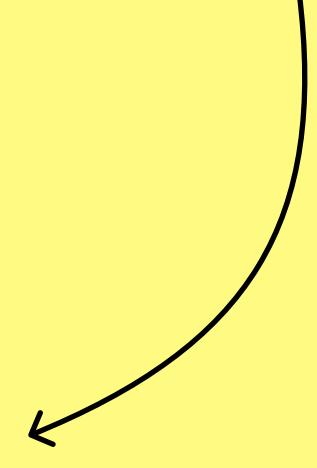


It enables developers to **build and manage APIs**.

# APIsec



APIsec is an **API security company**. It leverages automated testing tools to **find logic flaws** before your code hits the production stage.



APIsec **addresses the business need to secure APIs** before they reach production and provides the industry's only **automated and continuous** API testing platform that uncovers security vulnerabilities in APIs.

# Application

Application software is commonly defined as a **program** or a bundle of different programs **designed for end-users**.

Every **program** can be called an application, and often the terms are used interchangeably.

# Framework

A framework contains **libraries of code, instructions, and APIs** from which developers and API consumers can obtain information from an app.

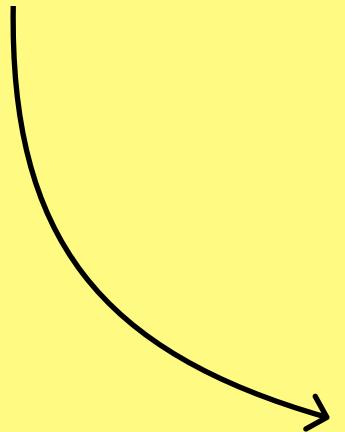
# Burp Suite

Burp -also called Burp Suite- is a **set of tools** used for **penetration testing** of web apps.

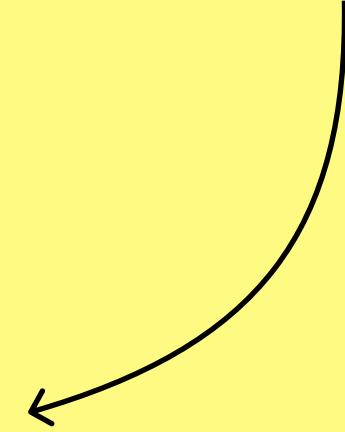
Burp is an **all-in-one penetration testing suite** that offers users a one-stop shop for all their pen testing needs.

BurpSuite contains an **intercepting proxy** that lets the user see and modify the contents of requests and responses while they are in transit for granular control of your APIs.

# CI/CD

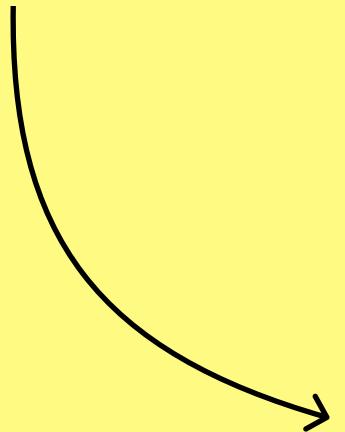


Continuous integration (CI) and continuous deployment (CD) are a set of **operating principles and a collection of practices** and agile methodologies that enable development teams to deliver better and faster changes to their code.

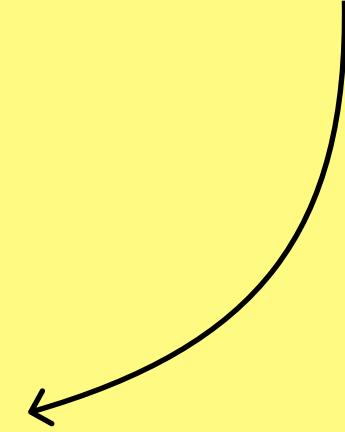


CI/CD is one of the most **important DevOps practices** as it gives teams the tools to focus on meeting their business requirements, code quality, and security needs.

# CRUD

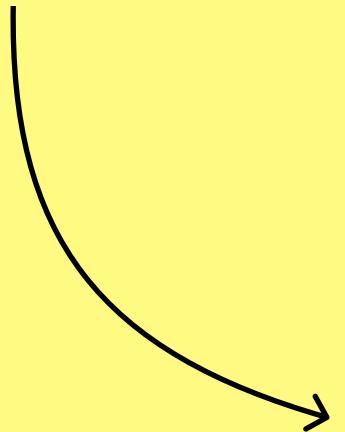


CRUD is an acronym for **create**, **read**, **update** and **delete**. It refers to the necessary functions to implement a storage application, such as a hard drive.



Unlike random access memory and internal caching, CRUD data is typically **stored and organized into a database**, which is simply a collection of data that can be viewed electronically.

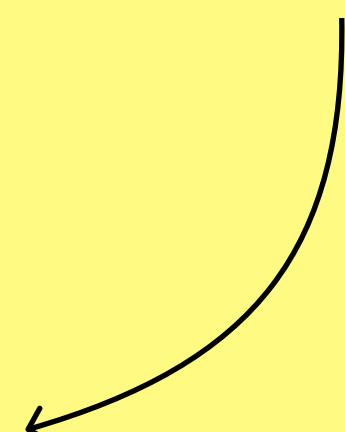
# Cache



The cache is a software or hardware component that stores data so users can access and retrieve that data faster. Cached data might be the result of a copy of certain data stored elsewhere.

Cache reads data and retrieves it faster than you would otherwise.

# Client



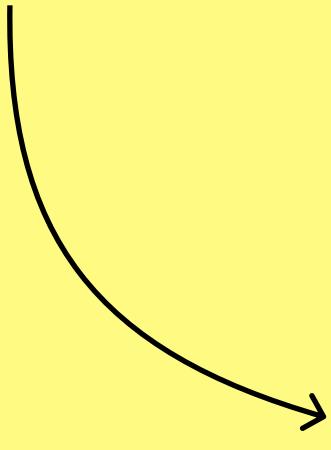
A client is a **device** that **communicates with a server**. A client can be a desktop computer, a laptop, a smartphone, or an IoT-powered device. Most networks allow communication between clients and servers as it flows through a **router or switch**.

# DDoS

A distributed denial of service (DDoS) attack is a **malicious attack** that aims at disrupting the target's traffic.

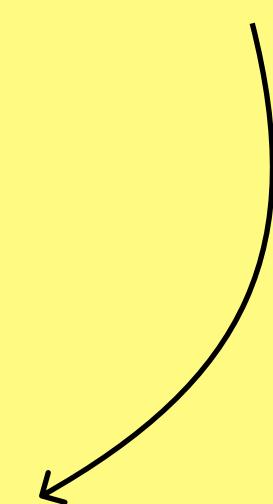
It usually overwhelms the target's infrastructure with a **flurry of internet traffic** aimed at saturating the servers and causing them to shut the page down.

# Resource



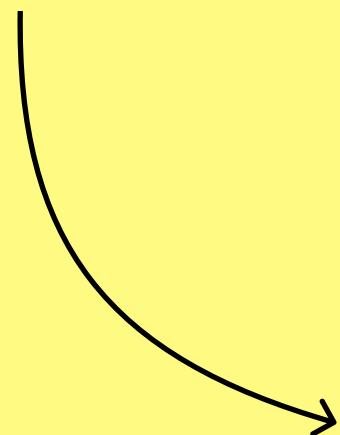
An entity that can be represented by a [URI](#) and can be accessed through an [API](#). **Resources** can be anything from [data](#) (such as a list of users or a single user's profile) to [operations](#) (such as creating or updating a resource).

# Request



An [HTTP request](#) sent by a client to a server to [retrieve](#) or [modify](#) [data](#). A request typically includes a method, a URI, and a set of headers and/or a body.

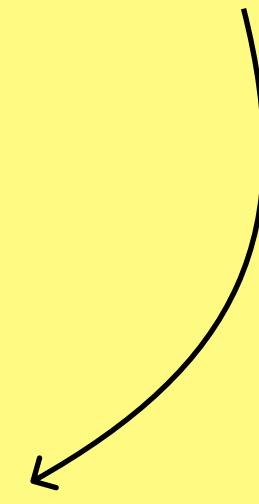
# Response



An **HTTP response** sent by a **server** to a **client** in response to a request.

# Response Code

A numerical status code returned in an API response to indicate the success or failure of a request. Common response codes include **200 (OK)**, **404 (Not Found)**, and **500 (Internal Server Error)**.



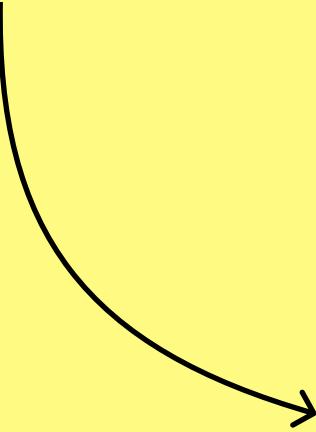
# Payload

The data sent in an API **request** or **response**, often in the form of a **JSON object**.

# Pagination

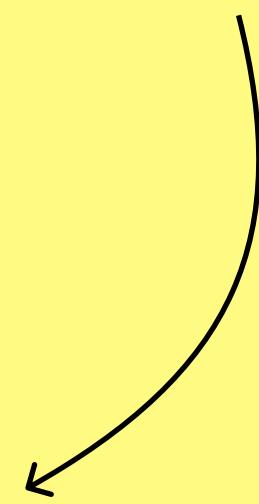
A technique used in APIs to divide a large dataset into smaller, more **manageable chunks** or **pages**. This allows a client to **request a specific page** of data rather than receiving the entire dataset all at once.

# Method



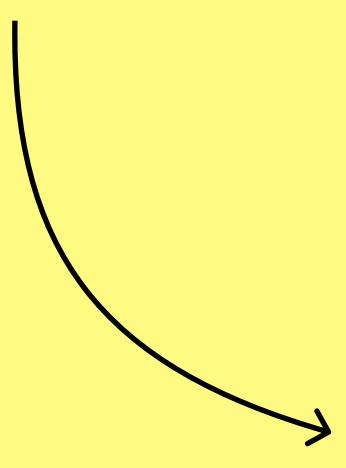
The **HTTP verb** used in an **API request**, such as GET, POST, PUT, or DELETE.

# Query Parameters



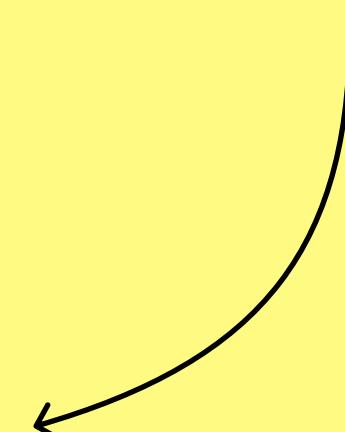
**Key-value pairs** that are added to the end of an **API endpoint URL** to specify certain criteria or **filters** for the data being requested.

# Authentication



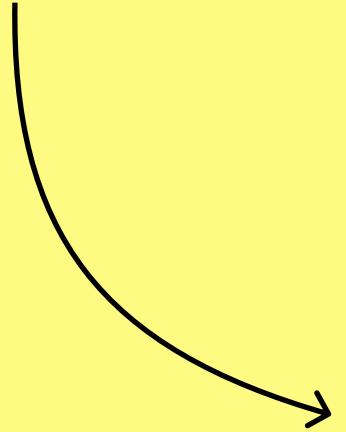
The process of **verifying the identity** of a client or user before allowing them to access an API. This is often done using an **API key** or other form of credentials.

# Rate Limiting



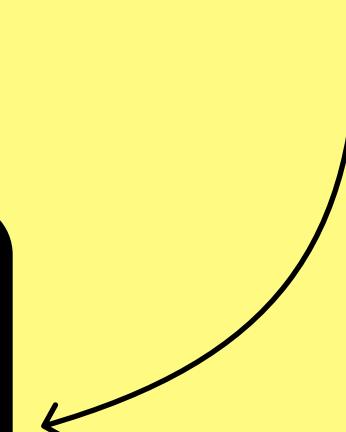
The process of **limiting the number of API requests** that a client can make within a certain **timeframe** to **prevent abuse or overuse** of the API.

# API Documentation



Detailed documentation or reference material provided by the creator of an API, explaining **how to use the API** and its various endpoints and parameters.

# Logic Flaw



Business logic flaws result from **faulty application logic**. In simple terms, a logic flaw happens when an application behaves unexpectedly. A logic flaw allows attackers to **misuse an application** and circumvent its rules to change how it performs.

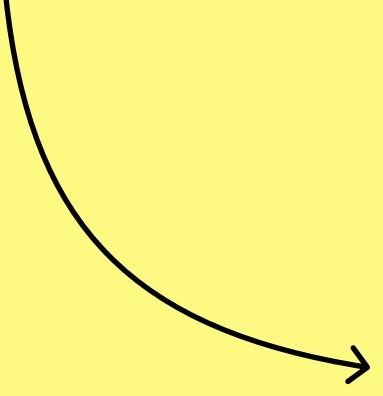
# JSON

JSON (JavaScript Object Notation) is a **lightweight data-interchange format** based on a subset of JavaScript programming language standards.

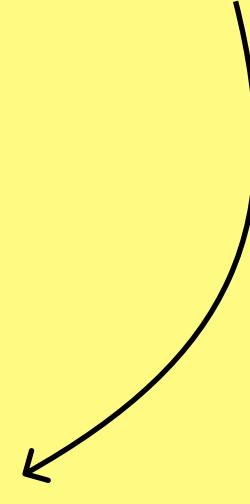
JSON has the advantage that it is both **easy for humans** to read and write **and for machines** to parse and generate.

It is a format that is completely **agnostic** to languages and uses conventions that are familiar to programmers of C-family languages.

# Microservices

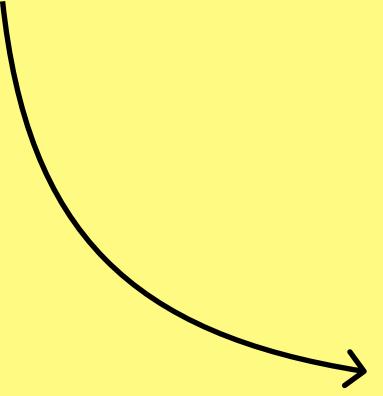


Microservices are also known as microservices architecture. It is a **software architecture style** that structures apps as a collection of **loosely coupled, independent**.

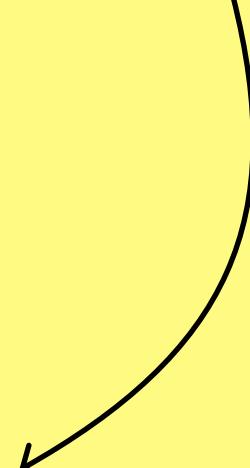


Microservices are highly maintainable services that are organized to **enhance** an app, website, or platform's **business capabilities**.

# Monetization

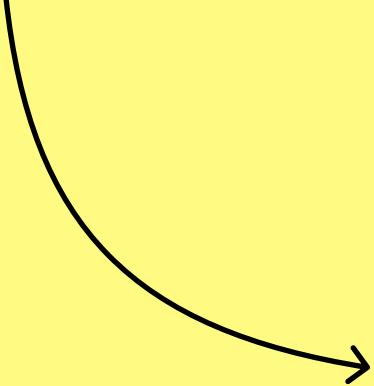


API monetization is a **process** by which a **business can create revenue** from its APIs.

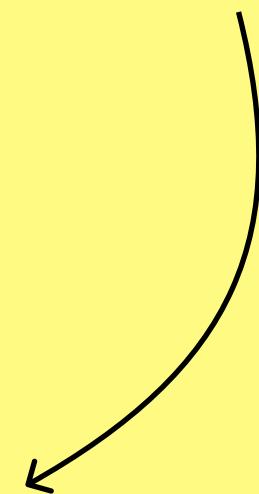


Since APIs enable users to **access and integrate data** from different sources, they can be used by different developers to **integrate relevant services** within their products, digital services, or applications, which could, in turn, become a **source of revenue** for both **public and private services** and applications.

# OWASP



OWASP (Open Web Application Security Project®) is a **nonprofit organization dedicated to enhancing software security.**



OWASP offers a range of tools to help developers and programmers secure the web through **open-source software projects**, hundreds of local chapters worldwide, and **educational and training events**.

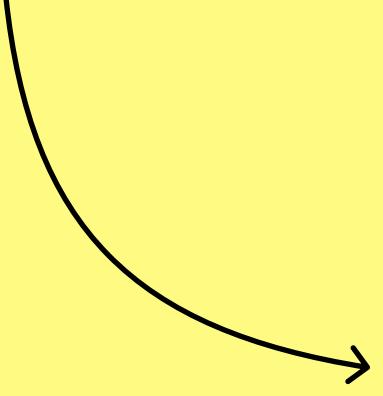
# Over-Permissioned Container

An over-permissioned container is a **container** that has all the **root capabilities** of a **host machine**.

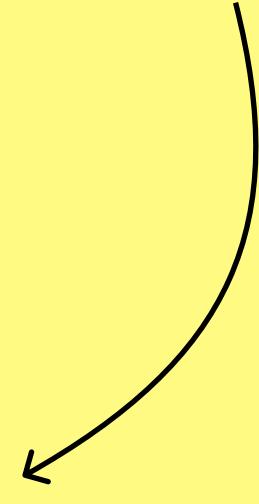
That means that it **can access resources** that aren't accessible to ordinary containers and users.

The problem with over-permissioning is that it gives **malicious actors** a point where they can **attack your infrastructure** and compromise your implementation.

# Parameters



Parameters are **special types of variables** used in computer programming to **pass information** between procedures and functions.



An **argument to a function** is referred to as a parameter. Adding three numbers, for example, may require three parameters.

# Penetration Testing

Also called **pen testing** or **ethical hacking**, penetration testing simulates attacks on your computer system to identify exploitable vulnerabilities.

Pen testing **identifies**, **tests**, and **highlights** vulnerabilities in an organization's security posture.

Web application firewalls (WAF) are generally augmented by penetration testing in the context of web application security.

# Production Environment

In a production environment, **software** and other products are **actually put into operation** in how their intended users intend them to be used.

Developers generally use this term to refer to the setting where **end-users** will actually **use the products**.

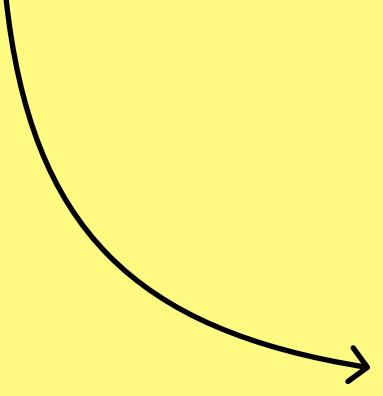
In a production environment, software programs and hardware are **run in real-time**, and they are relied on daily by organizations and companies for their daily operations.

# REST

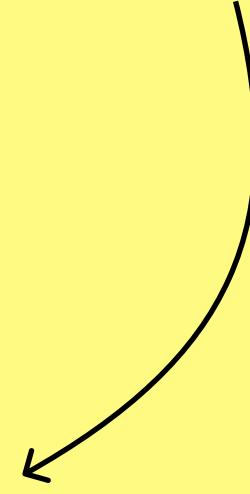
Created by [Roy Fielding](#), a computer scientist, REST, which stands for [REpresentational State Transfer](#), is an application programming interface that conforms to the constraints of [REST architectural style](#) and enables a quicker interaction between different [RESTful web services](#).

A [stateless Web service](#) must be able to read and modify its resources using a predefined set of operations and a textual representation.

# Red Teams

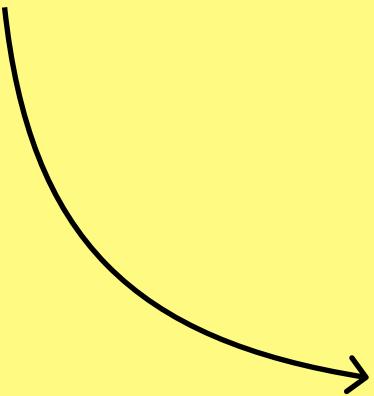


Red teams are **cybersecurity professionals** trained in attacking systems and breaking into them by finding **compromised entry points** or **exploitable logic flaws**.

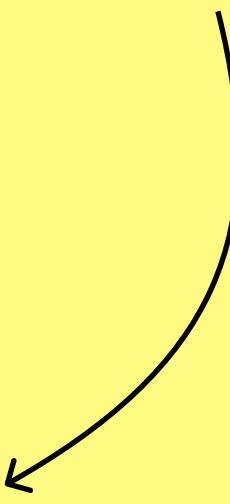


The objective of the red team is to **improve a company's cybersecurity standing** by showing it how they managed to gain access and exploit their **system vulnerabilities**.

# SDK



SDK stands for software development kit and is a **set of instructions, integrated practices, pieces, code samples, and documentation** that enables developers to create software applications on a specific software platform.



SDKs can be seen as **workshops** with everything developers need to **build specific software** for a determined **platform**.

# SDLC

SDLC -also called software development lifecycle- is the process for **planning**, **creating**, **testing**, and **deploying** an information system.

SDLC aims at producing **quality software** at the **lowest cost** in the **shortest time** possible.

SDLC gives developers a **structured flow** divided into **phases** to help companies produce high-quality software.

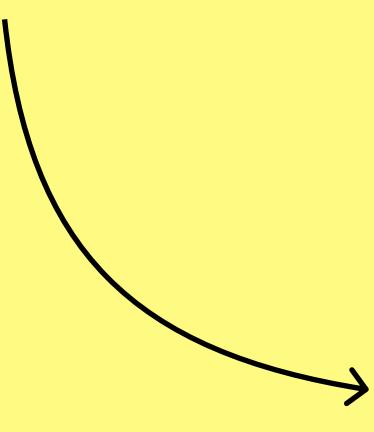
# SOAP

**Simple Object Access Protocol (SOAP)** is a protocol specification for **exchanging structured information** to implement web services.

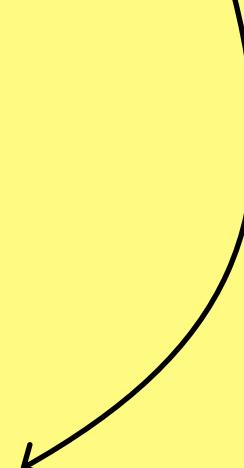
SOAP leverages **XML Information Set** for message format and other application-layer protocols, such as **HTTP** or **SMTP** for message transmission. The messaging services provided by SOAP are exclusively **XML-based**.

Microsoft originally developed the SOAP protocol to replace old technologies such as **Distributed Component Object Model (DCOM)** and **Common Object Request Broker Architecture (CORBA)** that cannot work over the internet.

# SQL Injection



An SQL injection technique is a way to **inject code** into a **database** that may damage it.



SQL injections are one of the most common web hacking techniques and rely on the placement of **malicious SQL code** in SQL statements via web input using forms or other editable fields.

# Webhook

A **webhook** (also called a web callback or HTTP push API) is a **way for an app to provide other applications with real-time information.**

Webhooks **deliver data directly** to other applications, so data is available immediately instead of standard APIs requiring frequent polling for **real-time data**.

Webhooks are **beneficial** to both consumers and providers in this way, but the only drawback is the **difficulty of setting them up at first.**

# ZAP

Also called [OWASP Zed Attack Proxy](#) (ZAP) is one of the world's most popular [free security tools](#), which lets you automatically [find security vulnerabilities](#) in your applications.

By automating penetration testing and security regression testing, [developers can automate](#) an application's security testing during the [CI/CD process](#).

With ZAP, you can also do nearly everything you can do with the desktop interface using its [powerful API](#).

# For More Interesting Content



## Brij Kishore Pandey



### Follow Me On LinkedIn

<https://www.linkedin.com/in/brijpandeyji/>