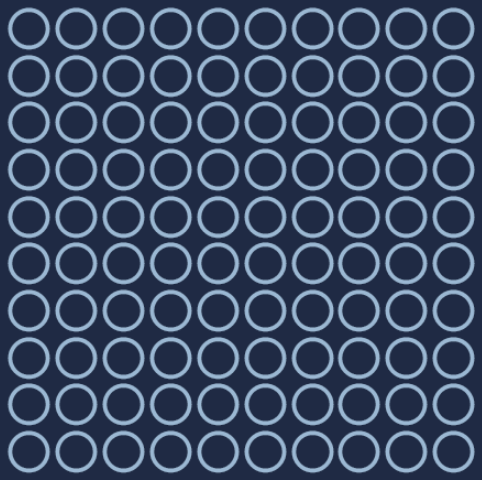
API Insecurities on the Rise

The number of external and internal application programming interfaces (APIs) that organizations are deploying is rising sharply as organizations deploy microservices-based applications in production environments with greater frequency. Each microservice within an application has its own API.

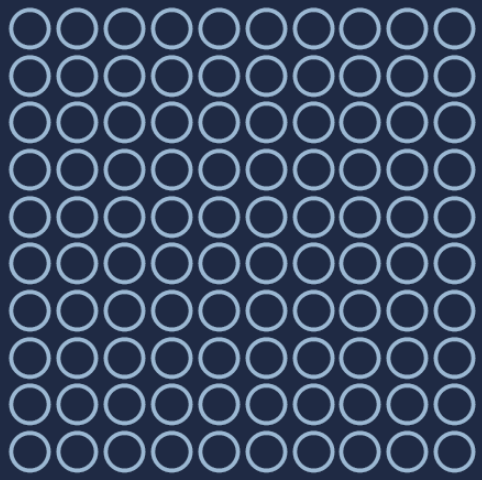
The challenge that creates is each of those API needs to be secured. In effect, the attack surface within IT environments increases with every API that provides access to some of an organization's most sensitive data. Add in all the APIs that developers are prone to misconfigure when provisioning infrastructure as code and the occasional "Zombie" API that someone forgets about deploying and it quickly becomes apparent API security is becoming a much larger issue by the day.

The challenge is finding a way to secure those APIs without unduly slowing down the rate at which applications are being deployed and updated. Unfortunately, it's not like developers are going to become API security experts overnight. More challenging still, APIs now increasingly come in formats such as REST, GraphQL or even SOAP. What's required is a level of cooperative effort between developers and cybersecurity experts within the context of a set of best DevSecOps practices that have thus far proven to be difficult to achieve and maintain.

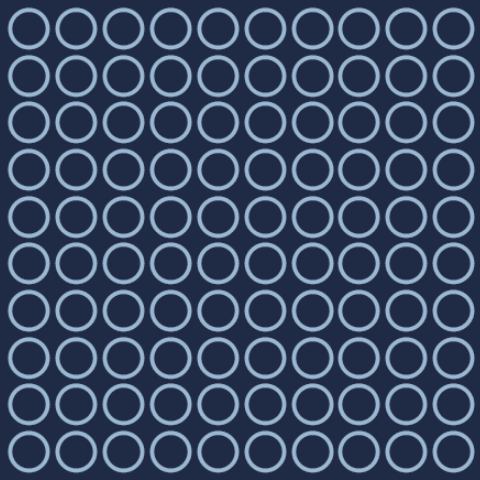
The State of API Security



Most organizations (94%) have experienced an API security incident in the past 12 months.



More than half have discovered a vulnerability in an API in the past 12 months.

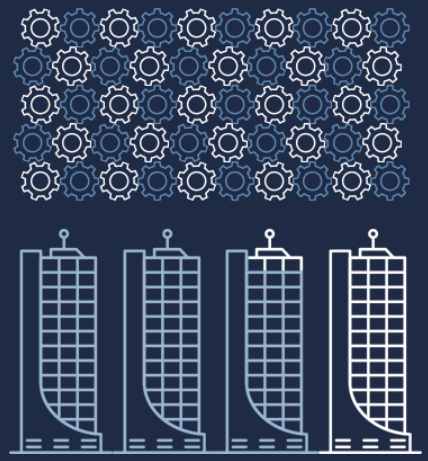


Nearly two-thirds (64%) of organizations have delayed application rollouts because of API security concerns.

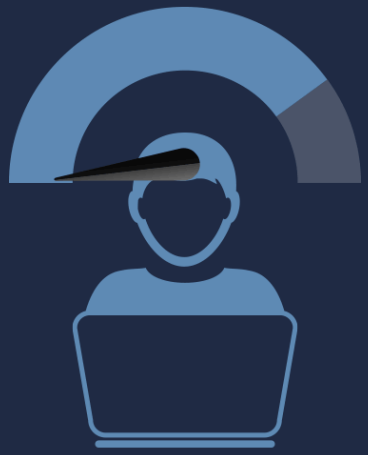
Source

API Security Becomes a Higher Priority

50+ APIs



Nearly **three-quarters of enterprises (73%)** use more than 50 APIs.



80% of security leaders would like to gain more control over their APIs.

Over the next 24 months, **91% of security leaders** will be making API security a priority.

Source

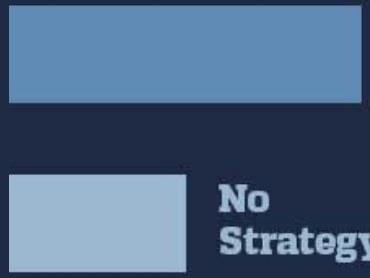
Some APIs Are More Trouble Than Others

A full **90% of companies** are vulnerable to security breaches due to cloud misconfigurations involving APIs.

More than 40% of users had at least one misconfigured Docker application programming interface (API) that, on average, takes 60 days to remediate.

Source

Limited Appreciation of API Security



More than half of organizations running APIs in production (54%) have, at best, only a basic strategy for API security, with 27% having no strategy at all.

Source

SECURITY
BOULEVARD

Top 10 API Security Issues

Identified by OWASP

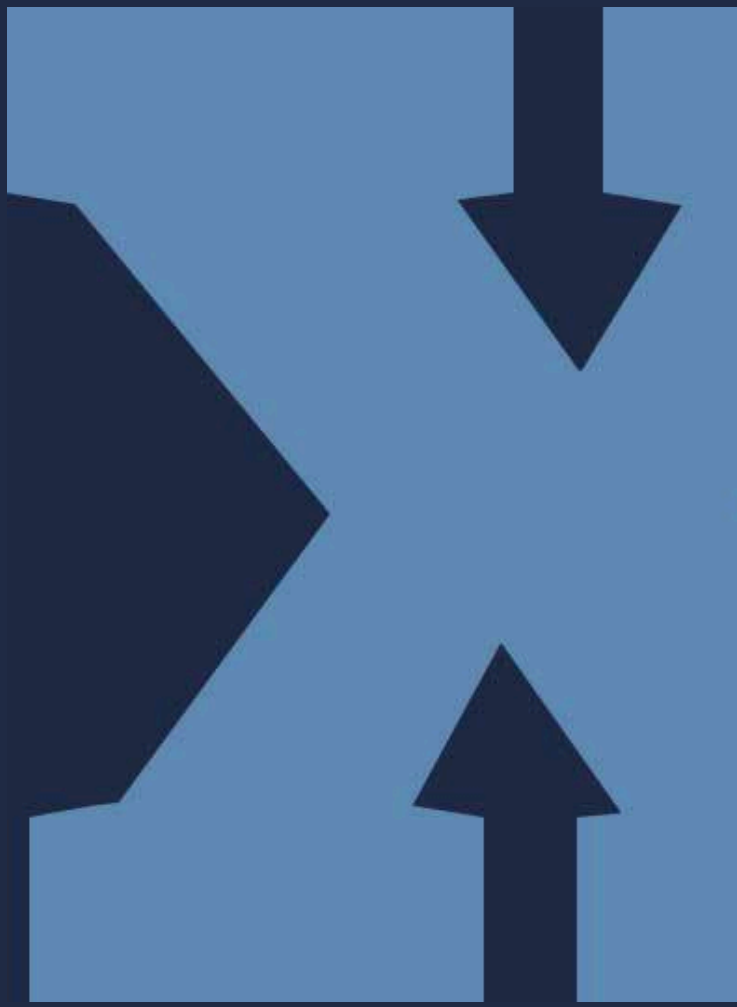
Broken Object Level Authorization: APIs expose endpoints that handle object identifiers that results in a wider attack surface. Object level authorization checks should be considered in every function that accesses a data source.

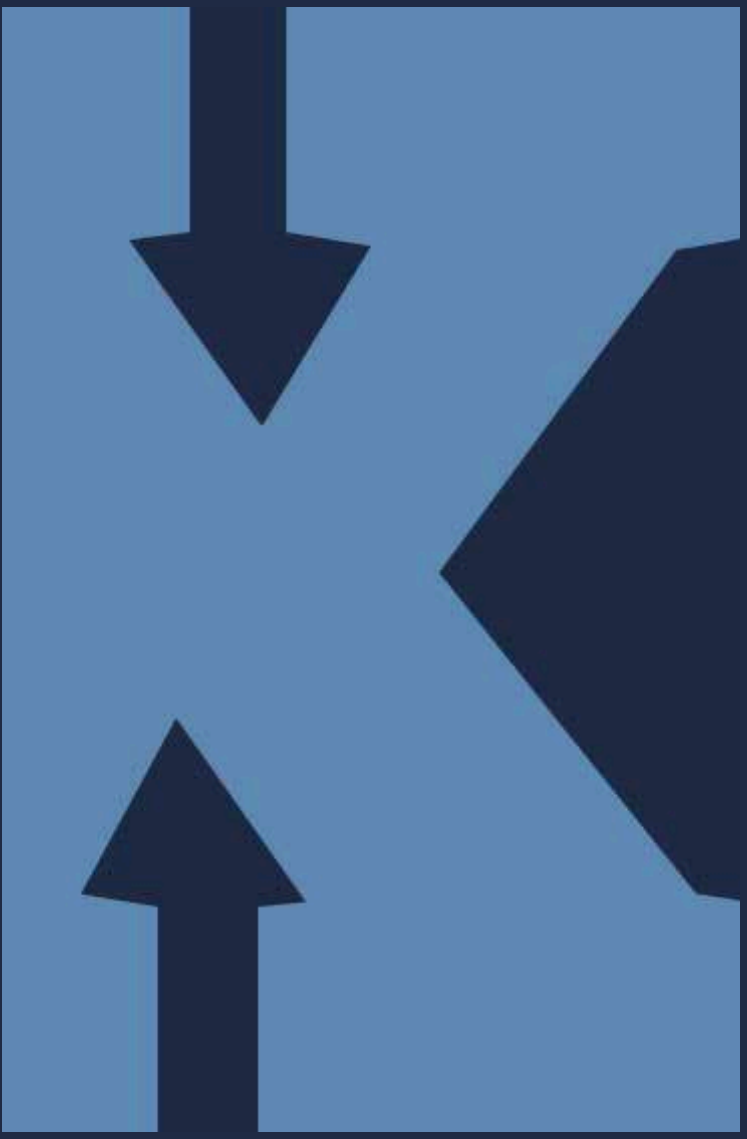
Broken User Authentication: Authentication mechanisms are often implemented incorrectly, allowing attackers to compromise authentication tokens or to exploit implementation flaws to assume other user's identities temporarily or permanently.

Excessive Data Exposure: Developers tend to expose all object properties without considering their individual sensitivity, relying on clients to perform the data filtering before displaying it to the user.

Lack of Resources and Rate Limiting: APIs often do not impose any restrictions on the size or number of resources that can be requested by the client/ user. That can lead to a denial of service (DoS) attack as well other types of brute force attacks against authentication protocols.

Broken Function Level Authorization: Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, create authorization vulnerabilities





Mass Assignment: Binding client provided data to data models, without proper properties filtering based on an allowlist allows attackers to modify object properties.

Security Misconfiguration: These commonly occur because of unsecure default configurations, incomplete or ad-hoc configurations, open cloud storage, misconfigured HTTP headers, unnecessary HTTP methods, permissive Cross-Origin resource sharing (CORS), and verbose error messages containing sensitive information.

Injection: These occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's malicious data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

Improper Assets Management: Documentation is critical to mitigate issues such as deprecated API versions and exposed debug endpoints.

Insufficient Logging and Monitoring: Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to launch additional attacks.

Source

Best API Security Practices

Testing APIs for vulnerabilities is now a critical element of any set of best DevSecOps practices as more responsibility for securing APIs continues to shift left toward developers. There's no shortage of types of tools for testing APIs. The most important thing is to make sure an API behaves regardless, as expected, of how it was built to ensure security.

Sponsored by:

STACKHAWK

www.stackhawk.com

IMVISION

www.imvision.ai

Thank you for reading

API Insecurities

Cookies [1] [2] Terms [1] [2] Privacy [1] [2]

POWERED BY Turtl