Whitepaper

The Cloud Native AppSec Playbook

Organizations across the globe have adopted advanced cloud architectures in recent years. Modern architecture has become increasingly more useful, bringing significant advantages to designing, building, and deploying applications in the cloud.

fastly

Table of Contents

Security Guidance for Multi-Cloud Environments

What you need to know

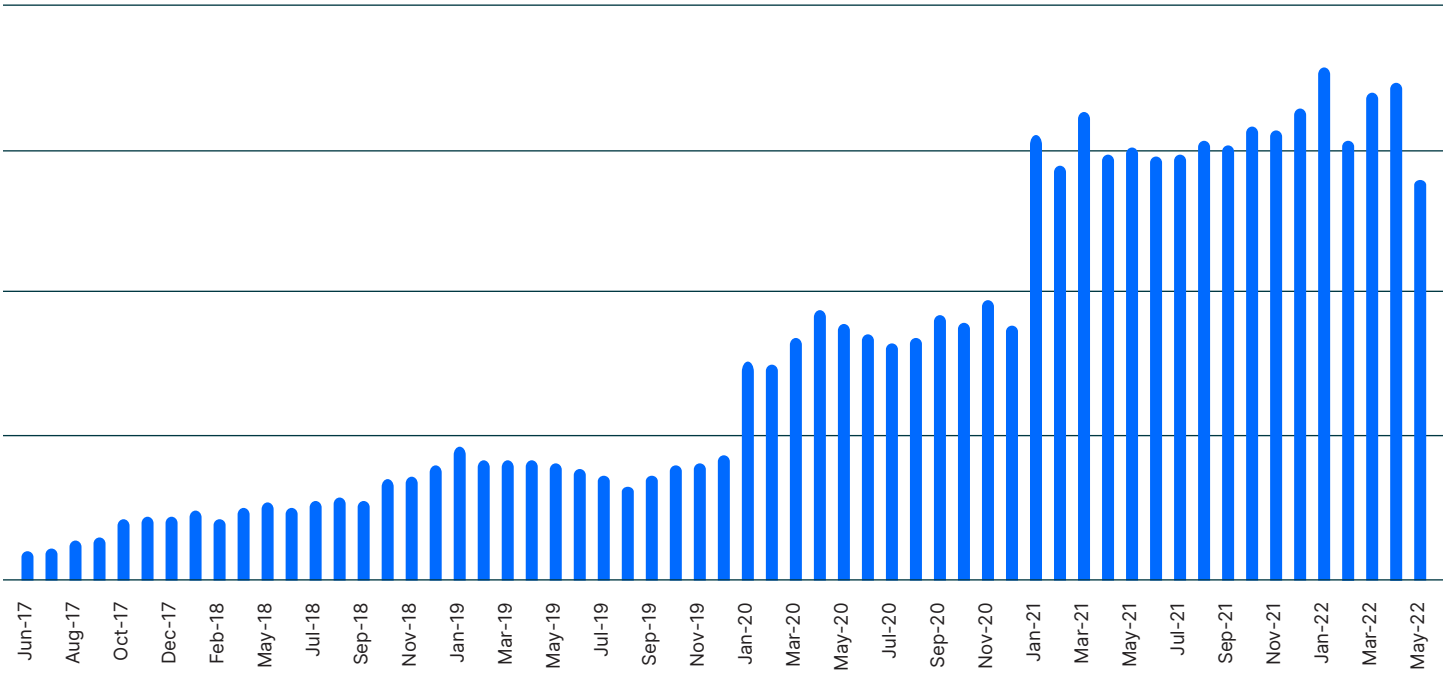
We leverage web applications to make key transactions that drive our daily personal and professional lives. From grocery shopping to telemedicine to submitting timesheets to retrieving digital paystubs, it's hard to imagine not using apps for everyday tasks. To deliver applications reliably, organizations increasingly use multi-cloud environments to scale and accelerate operations in order to serve expanding user bases. Understanding the unique challenges of running apps and APIs in the cloud is critical because users expect those apps to be available and performant.

Organizations across the globe have adopted advanced cloud architectures in recent years. Modern architecture has become increasingly more useful, bringing significant advantages to designing, building, and deploying applications in the cloud. IDC research says that "by 2023, over 500 million apps will be developed using cloud-native approaches."

The move to cloud is also driving container use. Gartner estimates that "90% of global organizations will be running containerized applications in production by 2026 – up from 40% in 2021. In addition, by 2026, 20% of all enterprise applications will run in containers – up from fewer than 10% in 2020."

From our observation and inspection of over three trillion web requests monthly from customers across all verticals, we've distilled years of data into a framework for cloud native application security. Since March 2020 our customers' web request traffic has increased by 100.2%. The reason for this jump is clear: web application and API usage has multiplied due to the impact of COVID-19 on our daily lives.

Web request volume protected per month



Defining cloud native

What is multi-cloud?

Multi-cloud means using two or more cloud computing services from different cloud vendors. Companies leverage multiple cloud providers to distribute the computing resources necessary to operate the various apps and APIs that drive their business. Utilizing multi-cloud environments also minimizes the risk of downtime and data loss.

Conversely, relying on one cloud provider increases the likelihood of negatively impacting all your customers should that cloud compute vendor suffer an outage or other infrastructure issue that takes your apps offline. Multi-cloud environments can be all-private, all-public or a combination of both.

Defining cloud native

Cloud native is a shift in writing software to interact with and take advantage of the features and APIs that cloud infrastructure provides. The cloud native approach to building apps purposefully leverages the benefits of the cloud delivery model: scalability, agility, observability, and resiliency. Cloud native is a style used when developing an application, not a location where an application runs. While the exact definition is ambiguous, there's an industry consensus that "cloud native" entails common elements like:

- **Container-based infrastructure**
- **Microservices architecture**
- **Serverless functions**

Defining “The Cloud”

Private cloud:

A cloud environment operated and maintained by an organization for its own private use.

Public cloud:

A cloud environment operated external to an organization. Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform are widely used public cloud services.

Hybrid cloud:

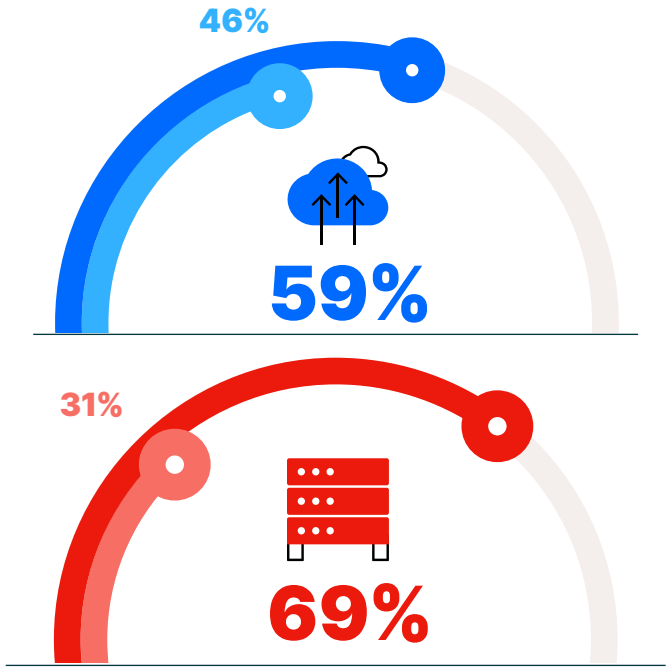
A mixture of cloud computing storage and services comprising on-premise, private cloud, and public cloud services.

Benefits & challenges of multi-cloud computing

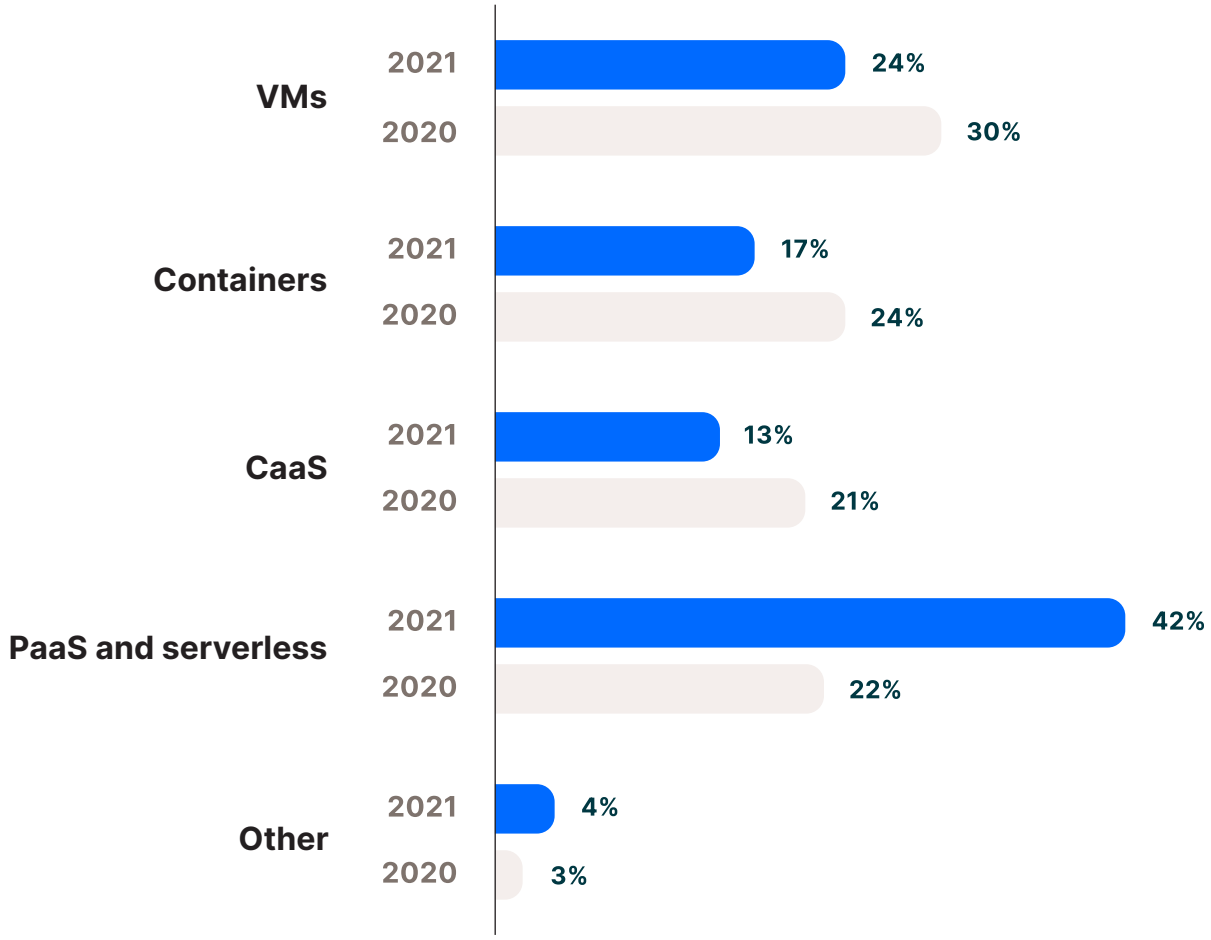
Cloud adoption through the pandemic

Throughout the pandemic, there were significant expansions of cloud workloads overall, jumping to an average of 59% of workloads hosted on the cloud, up from an average of 46% in 2020. In addition, 69% of organizations host more than half of their workloads in the cloud, up from just 31% of respondents in 2020.

While organizations continue to use diverse compute options, PaaS and serverless approaches rose 20 percentage points, while the use of containers and CaaS saw more moderate growth.



Percent change in cloud workload volumes since 2020.[1]



PaaS and serverless strategies, which allow development teams to put applications in the cloud without necessarily having to build and scale infrastructure at the same time, likely helped support the rapid transition to the cloud seen in the past year.

Benefits of cloud native: Agility & scalability

Cloud native environments promote speedy deployment, scalability, and flexibility, but with these benefits comes risk and blind spots. As DevOps teams accelerate their cloud native app development, major security concerns remain.

Top security challenges for cloud native applications

- **Real-time threat visibility**
  Organizations have limited real-time visibility into attacks: 73% say they lack actionable, fine-grain, real-time insight into threats and ongoing attacks.
- **False positives**
  False positives continue to plague IT and security teams: 46% of those surveyed said that more than half of production environment security alerts were false positives.
- **Insecure production environments**
  Production environments are not adequately protected: Nearly 50% surveyed saw damage inflicted on production systems and/or the loss of customer data. The challenges to securing apps and APIs operating in cloud-based environments are very real, but not insurmountable.

Reasons for expanding cloud capabilities

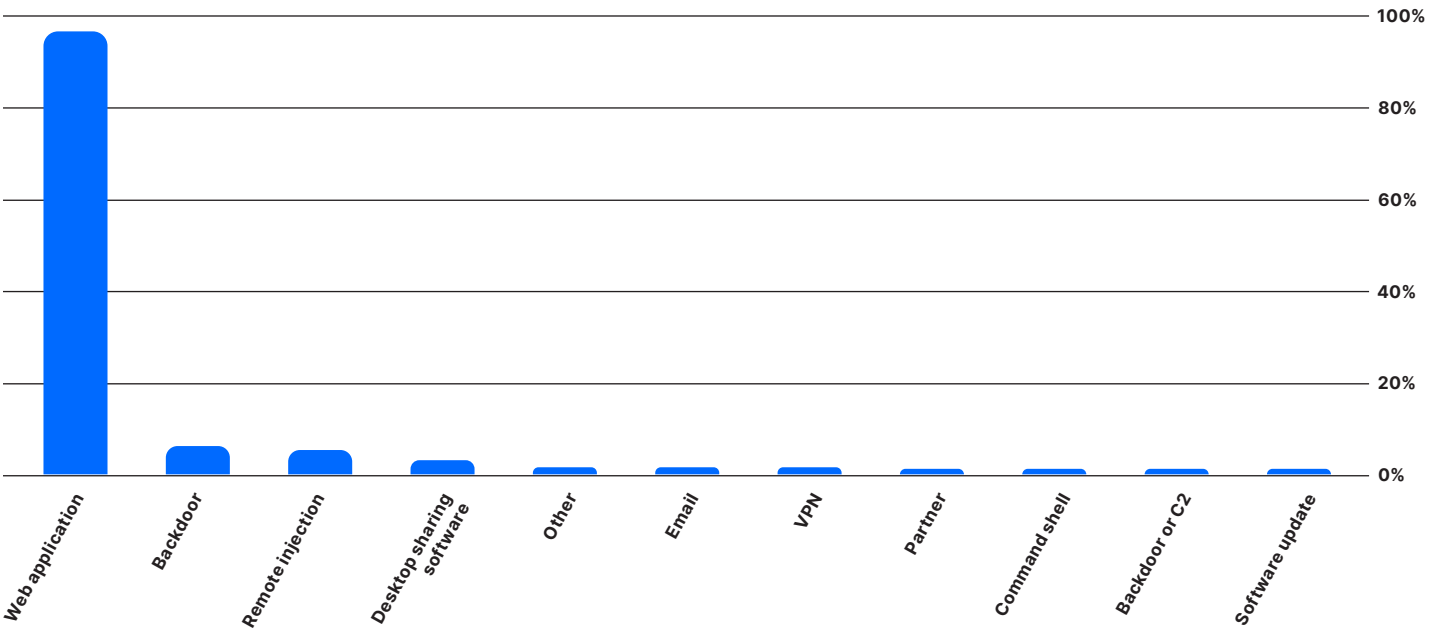
The flexibility and agility that the cloud provides to organizations allow them to keep businesses moving forward at an ever-increasing pace.

Reasons for expanding cloud capabilities.[1]

How attackers target cloud-based apps

Verizon's Data Breach Investigations (DBIR) 2022 report mentioned that web applications are the number one attack vector accounting for about 70% of security incidents. Furthermore, the same report stresses the importance of proper password protection since over 80% of Basic Web Application Attacks breaches can be attributed to stolen credentials.

Top Action vectors in Basic Web Application Attacks breaches (n=972)



The common attack techniques malicious threat actors employ when targeting applications are:

Web Attack Type	Description
Attack tooling	Using automated software to identify security vulnerabilities or exploit a discovered vulnerability.
Server side request forgery (SSRF)	Sending forged requests which appear to be legitimate to target internal systems.
Backdoor	Negates normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers.
Command execution	Using arbitrary system commands by means of user input to gain control or target a system.
Cross site scripting (XSS)	Hijacking a user's account or web-browsing session through malicious JavaScript code.
Directory traversal	Navigating privileged folders throughout a system in hopes of obtaining sensitive information.
SQL injection	Executing arbitrary database queries to find privileged information or gain access to an application.

APIs as a key attack target

APIs enable organizations to share data with authorized software developers and partner organizations who leverage that valuable data in their own applications. Distributed cloud native applications rely on APIs to communicate between microservices that perform key application functions. As a component of modern business innovation and software development, APIs enable applications to exchange data and, in effect, "talk to" one another. API security is critical to the defense of cloud native applications.

> Gartner estimates that by 2022, API abuses will be the most frequent attack vector for enterprise web applications data breaches[3].

API security threats

The OWASP Foundation publishes an API Security Top 10 list that practitioners and security stakeholders can reference. Below are the top API attack scenarios:

OWASP API Security Top 10	Attack Scenario
Broken Object Level Authorization	APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface Level Access Control issue.
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	Looking forward to generic implementations, 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 & Rate Limiting	Quite often, APIs do not impose any restrictions on the size or number of resources that can be requested by the client/user. Not only can this impact the API server performance, leading to Denial of Service (DoS), but also leaves the door open to authentication flaws such as brute force.
Broken Function Level Authorization	Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, tend to lead to authorization flaws. By exploiting these issues, attackers gain access to other users' resources and/or administrative functions.

Mass Assignment	Binding clients provided data (e.g., JSON) to data models, without proper properties filtering based on a whitelist, usually lead to Mass Assignment. Either guessing objects properties, exploring other API endpoints, reading the documentation, or providing additional object properties in request payloads, allows attackers to modify object properties they are not supposed to.
Security Misconfiguration	Security misconfiguration is commonly a result of insecure 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	Injection flaws, such as SQL, NoSQL, Command Injection, etc., 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	APIs tend to expose more endpoints than traditional web applications, making proper and updated documentation highly important. Proper hosts and deployed API versions inventory also play an important role to mitigate issues such as deprecated API versions and exposed debug endpoints.
Insufficient Logging & Monitoring	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems to tamper with, extract, or destroy data. Most breach studies demonstrate the time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

How to secure cloud native apps & APIs

The portability of containers is a fundamental reason that the traditional perimeter security framework is not sufficient in multi-cloud environments: containerized workloads can and will be deployed in a large variety of environments, including multi-cloud. Unlike traditional monolithic applications that had set rules on where and when they could be deployed, containerized apps are deployed often and quickly with few restrictions.

> In December 2019, Google outlined a new model for application security for cloud native applications — "BeyondProd" from its influential "BeyondCorp" approach to enterprise network security and implementation of a zero-trust network. The white paper asserts that microservices today are a lot like end-users: they are no longer tied to the same IP address day in, day out, but instead are highly mobile and yet have to be secure everywhere.

Cloud native application security requirements

The following requirements are prescribed for securing cloud native applications[3]:

1. Mutually authenticated service endpoints
2. Transport security
3. Edge termination with global load balancing and denial of service protection
4. End-to-end code provenance
5. Runtime sandboxing

Service meshes can handle the first two requirements. Istio provides mutual TLS (mTLS) encryption between microservices deployed on the service mesh, authenticating microservices and encrypting service-to-service (east-west) traffic. A service mesh handles authorization using controls that allow or deny communications between services.

Service mesh + next-gen WAF integrations

Web application firewalls (WAF) originated from a time when perimeter security was deemed sufficient. WAFs are still predominantly deployed at the perimeter (or edge) of an application. Is there a way for the layer 7 protection that WAFs provide to be deployed in a zero-trust model, ensuring that malicious internal requests sent between microservices (and not just external requests) are detected and blocked?

With traditional appliance-based WAFs this would be impossible. But newer software-based WAFs, such as the Fastly Next-Gen WAF (powered by Signal Sciences) with an agent-based architecture, can provide protection. Our customers routinely deploy our software-based next-gen WAF both at the perimeter and on individual workloads behind the firewall. This is made possible by our agent-module architecture.

Service meshes can inspect east-west traffic between microservices in cloud native applications. A WAF agent can integrate directly with an Envoy cloud native proxy and Istio service mesh, easily deploying the WAF throughout your cluster to ensure traffic inspection and malicious request blocking is enabled.

Cloud native application security defined

A cloud native appsec solution should provide key capabilities:

Inspects both east-west (service to service) and north-south (client to server/ app origin) traffic routed via microservices architectures without code changes

Increased flexibility to deploy Layer 7 protection in cloud native applications

Increases Layer 7 visibility with simplified deployment for containerized microservices orchestrated via Kubernetes

Holistic web security & visibility

The web-tiers of modern applications are often deployed in public cloud platforms for two main reasons:

1. Auto-scaling the back end to meet peak demand periods
2. Access to content delivery networks (CDNs) for local edge-based access to scale the front end

To achieve these performance and quality-of-service objectives, DevOps teams refactor their web apps or build them from scratch on cloud native architectures. Because exploits against known vulnerabilities and zero-day threats are a constant, a defense-in-depth approach is required to protect these apps from compromise.

A comprehensive web application & API protection (WAAP) platform

Cloud native applications require security that works seamlessly within existing environments and integrates natively with your current tech stack.

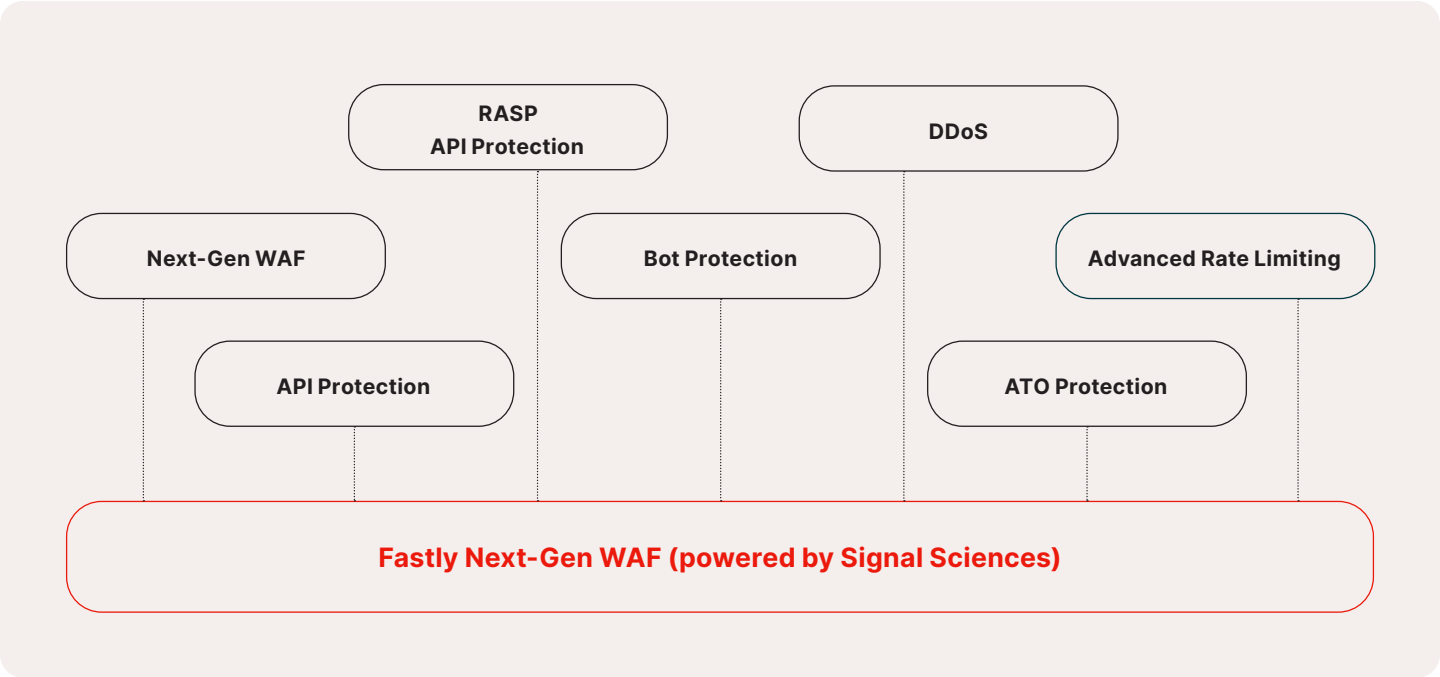
This reduces the amount of time and resources spent on systems integrations and sandboxing. Centralized management is also key to cloud security enforcement.

Gartner coined the term “web application and API protection”, shortened to “WAAP,” in 2017 in response to the evolution and expansion of web application firewall (WAF) capabilities.

A holistic web protection solution offers multiple protective capabilities that defend against both common and advanced web attacks and replaces multiple point products with different interfaces and lower defense efficacy.

In any environment, an integrated web protection platform provides comprehensive web security that defends against production cloud workload threats. WAAPs provide visibility and actionable information for your entire app footprint across multi-cloud environments.

A complete WAAP platform should deliver the following protective capabilities:



Key principles for cloud native appsec

Traditional perimeter solutions are insufficient to secure multi-cloud environments

Cloud native application security in multi-cloud environments means more than employing traditional perimeter- and rules-based solutions to mitigate injection-style attacks. DevOps and modern software delivery teams need newer, effective methods to protect cloud native applications against new attack methods in production cloud environments.

Cloud native apps require a unified security solution

Holistic web application security entails deploying a web application and API protection (WAAP) platform that includes a next-gen WAF as a key capability to combat a wide range of web attacks. A WAAP provides the proactive means to monitor web request traffic and instrument web requests before the payloads reach the app or API origin.

Flexible, scalable security = success for DevSecOps teams

A cloud native web defense solution also allows DevOps teams to use automated web defense with high precision in production, unlike legacy WAF vendors that rely on regular expression pattern matching rules and signatures.

Leveraging the scale and speed of the cloud doesn't mean security gets left behind. Legacy security tools weren't built to protect modern, cloud native applications. Our next-gen WAF puts DevOps and security first. Our technology protects apps in any infrastructure, integrates seamlessly into your DevOps and security tools, reduces friction, and increases collaboration between security and development teams.

Endnotes

1 The State of Cloud Native Security Report 2022. https://www.paloaltonetworks.com/state-of-cloud-native-security

2 “How to Build an Effective API Security Strategy | Gartner Research.” Accessed August 2020. https://www.gartner.com/doc/3834704

3 “BeyondProd: A new approach to cloud native security | Google.” Accessed December 2019. https://cloud.google.com/security/beyondprod