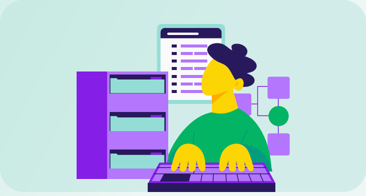
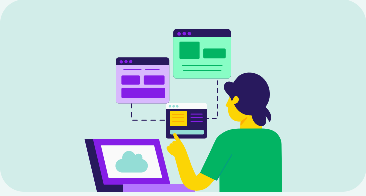
What is Runtime Security?

Have you ever come across the term “Runtime Security” and found yourself wondering what it really means? Does it refer to securing applications, protecting system operations, or reinforcing the entire architecture against different types of attacks? The truth is, the term “Runtime Security” can hold different definitions depending on the context and the perspective of various professionals:



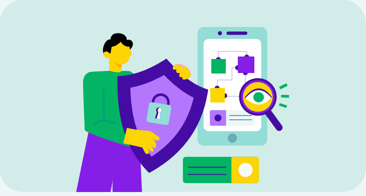
APP DEVELOPERS

"I consider runtime security as safeguarding an application during its execution, including runtime libraries, runtime behavior, and handling runtime errors. "



CLOUD ARCHITECT

"Runtime security is more about the operational aspect of applications within cloud environments. For example, our cloud-native apps consist of containerized runtime workloads that require monitoring and management within the application infrastructure. "



SECURITY ANALYST

"I see runtime security primarily as protecting applications and systems while they are actively running. This includes detecting and responding to threats that arise during the operation of applications, whether from external networks or within our own systems. "

Given this diversity of perspective, let’s define runtime security further. In general, runtime security is about protecting applications and systems while they are actively running and processing data. It involves establishing a baseline of normal behavior for each component in a cloud-native environment, including containers, hosts, and serverless functions. By continuously monitoring file systems, processes, and network activities, runtime security can quickly detect and respond to anomalies that may indicate a security threat. Runtime security is especially

important for organizations embracing modern apps with cloud-native architectures in dynamic, distributed environments (private, public and hybrid cloud), where traditional security measures lack visibility and are inadequate.

WHY IS RUNTIME SECURITY IMPORTANT NOW?

As organizations increasingly adopt modern applications with cloud-native architectures — whether in private, public, or hybrid clouds—and deploy technologies like VMs, containers, and serverless functions, runtime security becomes crucial for addressing real-time threats. Gartner projects a 19.9% compound annual growth rate (CAGR) through 2026 [1], highlighting the urgent need for robust runtime security measures.

Modern cloud environments are fast-paced and constantly changing, making runtime security essential for several important reasons:

Expanding Attack Surface	As organizations continue to migrate to the cloud, the attack surface "expands exponentially," leading to an increased number of potential vulnerabilities. CrowdStrike reports that "cyberattacks conducted by cloud-savvy threat actors—those who gain access to a victim's cloud environment and exploit cloud services— increased by 110% in 2023" [6]. Similarly, Check Point Research noted a "48% increase in cloud-based attacks in 2022" [4]. This growing risk underscores the importance of runtime security in "identifying and addressing vulnerabilities in running applications and workloads," hence protecting against these evolving threats.
Shared Responsibility for Security	Cloud security operates under a shared responsibility model, where "cloud providers secure the infrastructure while customers are responsible for protecting their applications and data." Runtime security is essential for helping customers fulfill their part of this model. According to a Microsoft Security Intelligence Report, "73% of organizations experienced a security incident due to confusion over this shared responsibility" [2]. To prevent such incidents, it is crucial to "proactively implement strong runtime security measures."
Complex & Dynamic Infrastructure	The cloud's inherently dynamic nature, often involving multiple services and providers, "creates a complex security landscape." This complexity demands continuous protection and visibility into running applications. Runtime security enables "real-time threat detection and response," ensuring that applications remain secure as they dynamically interact with various cloud services. Gartner predicts that by 2025, "99% of cloud security failures will be attributed to customer mismanagement," further emphasizing the need for robust runtime security solutions [3]."

RUNTIME RISKS IN MODERN ARCHITECTURES

In the evolving landscape of cloud-native architectures, understanding the importance of runtime security is critical. As organizations adopt advanced technologies such as containers, orchestrators, and serverless platforms, these components become attractive targets for cyber attackers. Identifying and mitigating the vulnerabilities within these modern runtime environments is essential to safeguarding applications and maintaining overall system security.

CONTAINER VULNERABILITIES:

Containers are a fundamental part of modern architectures, but they come with their own set of risks:

CONTAINER ESCAPE

This occurs when an attacker breaks out of a container and gains root access to the host system. An example is the CVE-2019-5736 vulnerability in runc, a popular container runtime.

IMAGE-BASED VULNERABILITIES

These arise when an application uses outdated or insecure base images. Attackers might exploit known vulnerabilities in popular Docker images from public repositories.

MALICIOUS IMAGE INJECTION

Attackers might insert malware or backdoors into container images, compromising the system. Another threat is crypto jacking, where attackers inject mining software into containers, by exploiting an exposed Docker remote API.

ORCHESTRATOR VULNERABILITIES (E.G., KUBERNETES)

Orchestrators like Kubernetes are powerful tools for managing containerized environments, but they are also targets for attackers:

API SERVER EXPLOITATION

Attackers could gain unauthorized access to cluster-wide resources, as seen in the CVE-2018-1002105 Kubernetes privilege escalation flaw.

KUBELET EXPLOITATION

If the Kubelet API is exposed, unauthorized parties can retrieve information or execute commands within containers. For example, anyone with access to the Kubelet service port (10250) can execute commands inside containers without needing a certificate.

MISCONFIGURATIONS

Misconfigured orchestrators can pose significant vulnerabilities. For instance, if the Kubernetes dashboard is exposed without authentication, attackers could take over the entire cluster. Similarly, running containers with root privileges or mounting sensitive host paths are other common security risks.

SERVERLESS PLATFORM VULNERABILITIES

Serverless architectures are increasingly popular due to their scalability and efficiency, but they also introduce unique security challenges:

INSECURE API GATEWAYS

Misconfigured API gateways can lead to unauthorized access or data exposure. For example, improper rate limiting could allow Denial of Service (DoS) attacks, or APIs might return excessive data, such as full user records, instead of only the necessary information.

INJECTION ATTACKS

APIs in serverless architectures are particularly vulnerable to injection attacks if input validation is not properly implemented. For example, a NoSQL injection attack could target a MongoDB-backed serverless API.

BUILDING A ROBUST RUNTIME SECURITY

STRATEGY: Security measures have evolved from focusing solely on simple web applications to encompassing APIs, cloud workloads, and entire cloud environments. This transformation requires that security approaches adapt to support these recent technologies. To meet these challenges, modern businesses must develop a strong, multi-layered runtime security strategy.

Organizations typically achieve this by implementing a combination of technologies tailored to their specific needs and application architecture. Key components of this strategy include Web Application Firewalls (WAF), Runtime Application Self-Protection (RASP), Web Application and API Protection (WAAP), and Cloud-Native Application Protection Platforms (CNAPP).

We will now explore how these technologies provide protection in diverse scenarios:

WEB APPLICATION FIREWALLS (WAF)

A Web Application Firewall (WAF) is a security solution deployed at the network layer to safeguard an organization's web applications. It was one of the first lines of defense against web-based attacks, designed to filter, monitor, and analyze HTTP and HTTPS traffic between a web application and the internet.

WAFS EMPLOY A COMBINATION OF TECHNIQUES TO IDENTIFY AND MITIGATE THREATS:

SIGNATURE BASED DETECTION

WAFs maintain a database of known threat signatures, enabling them to detect and block malicious activities based on these predefined patterns.

ANOMALY DETECTION

WAFs use anomaly detection to spot unusual patterns in web requests and responses that could signify an attack.

LIMITATIONS

The primary objective of a WAF is to provide broad protection for all an organization's internet- facing web applications. While WAFs can be configured to offer specific protections for different applications, their decisions are based solely on the data visible in network traffic. Therefore, WAFs may not catch certain threats that are not evident from the traffic alone.

RUNTIME APPLICATION SELF-PROTECTION (RASP)

Runtime Application Self-Protection (RASP), a term coined by Gartner in 2012, is a security technology designed to offer targeted protection for individual applications. Unlike Web Application Firewalls (WAFs), which secure multiple applications by monitoring network traffic, RASP focuses on the internal behavior and data flow of a single application. It does this by embedding security controls directly within the application itself.

KEY FEATURES OF RASP:

Introspection

RASP conducts detailed monitoring of an application’s inputs, outputs, and behavior.
This enables it to detect and respond to threats based on their impact on the application, rather than relying on predefined patterns.

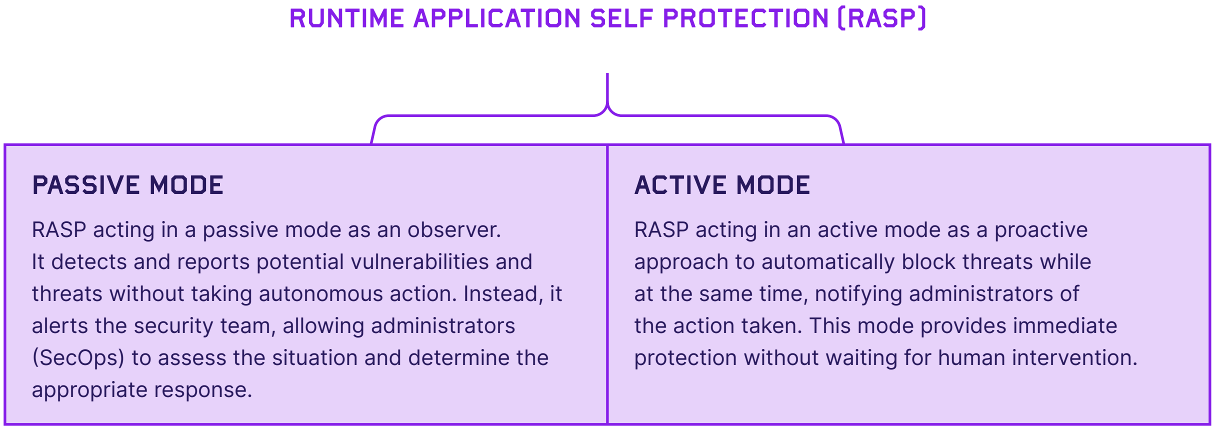
Contextual Analysis

RASP analyzes the application’s runtime context to determine whether the application is performing as expected. By operating on the same server as the application, RASP can detect and block attacks immediately, thus identifying sophisticated threats that might evade traditional security measures.

RASP operates on the application’s server, automatically activating when the application starts. Since RASP is embedded within the application, there is no need for manual activation by an administrator. Upon detecting a potential threat, RASP can take actions such as terminating the session of the suspected malicious actor, sending a warning to the administrator, or shutting down the application. To identify threats, RASP uses methods such as pattern matching, threat analysis, data monitoring, and language security techniques.

RASP FUNCTIONS IN 2 OPERATIONAL MODES

Runtime Application Self Protection(RASP)



RUNTIME APPLICATION SELF PROTECTION (RASP)

PASSIVE MODE

RASP acting in a passive mode as an observer.
It detects and reports potential vulnerabilities and threats without taking autonomous action. Instead, it alerts the security team, allowing administrators (SecOps) to assess the situation and determine the appropriate response.

ACTIVE MODE

RASP acting in an active mode as a proactive approach to automatically block threats while at the same time, notifying administrators of the action taken. This mode provides immediate protection without waiting for human intervention.

LIMITATIONS

Each of these analytical methods has its strengths and weaknesses. For example, some methods may be prone to false positives (incorrectly identifying activities as threats), while others might result in false negatives (failing to detect actual threats).

WEB APPLICATION & API PROTECTION (WAAP)

According to Gartner, cloud WAAP (Web Application and API Protection) is a comprehensive set of security tools designed to protect web applications, no matter where they are hosted. Cloud WAAP covers key security risks, including the OWASP Top 10 web application vulnerabilities and automated attacks. It also provides robust API security and can detect and defend against complex Layer 7 attacks that target web applications. The core components of Cloud WAAP work together to create a holistic security approach for web-based applications in cloud environments. These components include:

- **Web Application Firewall (WAF)**
- **Bot Management Systems**
- **Distributed Denial of Service (DDoS) Mitigation Tools**
- **API Protection Mechanisms**

LIMITATIONS

WAAP offers extensive protection against a wide range of application layer attacks, including bot mitigation, DDoS protection, and API security. However, organizations may face several challenges when implementing and maintaining WAAP solutions:

POTENTIAL LATENCY

WAAP solutions may introduce latency, which could impact application performance and user experience—critical factors for businesses that rely on fast, responsive web applications and APIs.

FALSE POSITIVES

Despite using advanced machine learning techniques, WAAP solutions can still generate false positives, potentially blocking legitimate traffic. Managing this requires continuous tuning, which can be resource-intensive and may affect user interactions if not handled properly.

COMPLEX CONFIGURATION & MANAGEMENT:

The broad feature set of WAAP solutions can make them difficult to configure and manage, often requiring specialized skills and resources. This complexity can lead to misconfigurations or underutilization of features.

COST

Comprehensive WAAP solutions, particularly those with advanced features, can be expensive. This might make them inaccessible for smaller organizations or strain the budgets of larger ones.

CUSTOMIZATION NEEDS

WAAP solutions may not always provide adequate protection for highly customized or unique applications without significant customization. This can leave industry- specific applications vulnerable to targeted attacks.

Despite these limitations, WAAP solutions are continuously evolving, with a growing list of features, including:

ADVANCED ANALYTICS & REPORTING

Offering more sophisticated analytics and reporting features, WAAP solutions provide deeper insights into application traffic patterns and potential security risks.

ZERO TRUST INTEGRATION

Many WAAP platforms are now designed to integrate with Zero Trust architectures, supporting continuous authentication and authorization at the application layer.

SERVERLESS AND CONTAINER PROTECTION

As cloud-native applications become more prevalent, WAAP solutions are expanding to offer specific protections for serverless functions and containerized applications.

BOT MANAGEMENT

WAAP solutions now include advanced bot detection and mitigation, addressing increasingly complex bot-driven attacks such as credential stuffing, account takeover attempts, and web scraping.

API DISCOVERY AND PROTECTION

With the proliferation of APIs, WAAP platforms now feature automatic API discovery and schema validation, helping organizations protect even unknown or shadow APIs.

CLIENT-SIDE PROTECTION

To counter the growing threat of client-side attacks like Magecart, WAAP platforms are introducing features to monitor and protect against malicious code injections on the client side.

CLOUD-NATIVE APPLICATION PROTECTION PLATFORM (CNAPP)

The Cloud-Native Application Protection Platform (CNAPP) is a comprehensive, all-in-one security solution specifically designed for cloud-native applications and infrastructure. Introduced by Gartner in 2021, CNAPP integrates various security and compliance capabilities into a unified platform. Its purpose is to prevent, detect, and respond to the wide range of threats that cloud environments face, providing a robust security framework for organizations operating in the cloud.

Core Technologies Integrated Into Cnapp:

CLOUD SECURITY POSTURE MANAGEMENT (CSPM)

Monitors, alerts, and remediates misconfigurations and compliance risks within cloud environments.

CLOUD WORKLOAD PROTECTION PLATFORMS (CWPP)

Provides visibility and control over physical servers, virtual machines (VMs), containers, and serverless workloads across both cloud and data center environments, offering comprehensive protection for all types of workloads.

CLOUD INFRASTRUCTURE ENTITLEMENT MANAGEMENT (CIEM)

Mitigates the risk of data breaches by continuously monitoring identities, permissions, privileges, and activities within cloud environments, ensuring that access is appropriately managed and controlled.

IDENTITY & ACCESS MANAGEMENT (IAM)

Manages and controls access to internal resources, ensuring that users have the correct permissions and that access policies are enforced consistently across the cloud infrastructure.

DATA PROTECTION

Monitors, inspects, and prevents the exfiltration of sensitive data caused by malicious insiders, phishing, and other threats, protecting the integrity and confidentiality of critical information.

CNAPP brings together these diverse technologies into a single, unified platform that manages security across all cloud-native environments and applications. We will now explore how these technologies provide protection in diverse scenarios:

BENEFITS OF CNAPP

UNIFIED SECURITY MANAGEMENT

By integrating multiple security functions into a single platform, CNAPP simplifies the management of security across cloud environments, making it easier for organizations to maintain a robust security posture.

PROACTIVE THREAT DETECTION

CNAPP helps organizations stay ahead of evolving threats by proactively identifying and mitigating security risks across the cloud infrastructure, ensuring ongoing compliance with industry standards and regulations.

COST EFFICIENCY

The consolidation of security tools into a single platform can lead to significant cost savings, reducing the need for multiple licenses, maintenance costs, and operational overhead.

LIMITATIONS

Despite its many benefits, CNAPP also presents several challenges

STEEP LEARNING CURVE

Implementing and managing a CNAPP solution may require significant training for staff, given the platform's complexity. This can be a barrier for organizations with limited resources.

INTEGRATION DIFFICULTIES

Organizations may face challenges when integrating CNAPP with existing security tools and processes, potentially leading to operational disruptions.

ALERT OVERLOAD

The comprehensive nature of CNAPP can result in a flood of alerts, some of which may lack sufficient context, complicating the work of security operations (SecOps) teams.

VENDOR LOCK-IN

Relying on a single CNAPP solution for multiple security functions could lead to dependency on one vendor, increasing the risk of vendor lock-in.

PERFORMANCE IMPACT:

Extensive monitoring and protection offered by CNAPP might negatively affect application performance if not properly optimized, potentially impacting user experience.

THE EVOLUTION FROM WAF & RASP TO WEB APPLICATION & API PROTECTION (WAAP)

WAF and RASP are complementary security solutions. WAF serves as the first line of defense, filtering out many common threats before they reach the application. Meanwhile, RASP adds an additional layer of defense by identifying and blocking threats that bypass the WAF. The combined capabilities of WAF and RASP have evolved into the next generation of security technology known as Web Application and API Protection (WAAP).

THE EVOLUTION OF WAAP AND THE RISE OF CNAPP

As the threat landscape continues to evolve, WAAP solutions are expected to expand their capabilities to address new and emerging risks. The ongoing shift to cloud-native environments is also pushing the development of more cloud-centric approaches to application security. This focus on the complexities of modern cloud-native application environments has introduced a new security solution: the Cloud-Native Application Protection Platform (CNAPP).

TIMELINE OF SECURITY TECHNOLOGIES

The table below reflects the evolution of application architectures and the changing threat landscape:

Technology	Emerged	Initial Focus	Evolution	Key Capabilites
Web Application Firewall (WAF)	Late 1990s — to Early 2000s	Protecting traditional web applications	Hardware appliances to cloud-based solutions	HTTP Traffic filtering, protection against common web exploits
Runtime Application Self-Protection (RASP)	Early 2010s	Addressing WAF limitations with application-level protection	Improved integration with modern development & cloud	Real-time attack detection & prevention, context-aware protection
Web Application & API Protection (WAAP)	Mid to Late 2010s	Extending WAF capabilities to APIs & modern web apps	Incorporation of ML & advanced threat intelligence	Comprehensive app & API protection, bot mitigation, DDoS protection
Cloud Workload Protection Platform (CWPP)	Mid 2010s	Securing cloud-based workloads (VMs, containers)	Expanded to serverless & cloud-native technologies	Workload-centric security, vulnerability management, compliance
Cloud Security Posture Management (CSPM)	Late 2010s	Managing security posture across cloud environments	Integratuon with other cloud security tools	Configuration managment, c ompliance monitoring, risk assessment
Cloud-Native Application Protection Platform (CNAPP)	Around 2020 (Named 2021)	Combining CWPP & CSPM or comphrensive cloud-native security	Ongoing evolution, increasing DevSecOps integration	Full-stack cloud-native security, integrated risk management

Key Takeaways

Runtime security is essential for safeguarding applications at the most critical times—when they are actively running, serving users, and processing data. Unlike traditional security measures that focus on perimeter defenses, runtime security provides continuous protection during the actual operation of applications and services. As technology evolves, runtime security has expanded to include additional layers to safeguard new technologies, encompassing applications, containers, hosts, and serverless functions. This comprehensive approach leverages technologies like Runtime Application Self-Protection (RASP), Web Application and API Protection (WAAP), and Cloud-Native Application Protection Platforms (CNAPP).

As cloud environments become increasingly complex and the threat landscape continues to evolve, runtime security is essential for safeguarding the critical business processes that run within these environments. By providing real-time protection, runtime security helps organizations proactively defend against emerging threats.

References: [1] Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $600 Billion in 2023 [2] Microsoft Security Intelligence Report [3] Gartner: Is the Cloud Secure? [4] Check Point Research: Cloud Security Report 2022 [5] IBM Cost of a Data Breach Report 2022 [6] CrowdStrike's Global Threat Report.