Guide

Runtime Security: Detecting Threats in Real Time

uptycs

Table of Contents

Key Takeaways

- Runtime security monitors live workloads to detect and respond to active threats as they occur
- Posture management and vulnerability scanning have no visibility into attacks that happen after deployment
- Runtime telemetry surfaces process activity, network behavior, and workload anomalies in real time
- Runtime security exposes threats that evade static controls, including credential abuse, malicious processes, and lateral movement
- Uptycs traces runtime detections back to the originating code and configuration changes, connecting security signals from code to cloud

Runtime security focuses on monitoring, detecting, and responding to threats while applications and workloads are actively running. Unlike build-time scanning or posture checks, it provides visibility into what is actually happening inside your environment as workloads execute.

As cloud environments become more dynamic with containers, Kubernetes, serverless, and ephemeral infrastructure, traditional security controls struggle to keep up. Misconfigurations can be fixed and vulnerabilities can be patched, but attacks can still succeed after deployment through credential theft, runtime exploitation, and zero-day vulnerabilities.

Runtime security helps security teams understand real behavior across workloads, making runtime security essential for detecting and responding to threats with context.

What Is Runtime Security?

Runtime security is the practice of protecting workloads during execution by continuously monitoring system activity, detecting threats, and enabling response.

Unlike static security approaches that rely on periodic scans or posture management and vulnerability scanning, runtime security focuses on what is happening in real time inside active environments. It helps organizations understand how workloads behave after deployment, where attacks actually occur.

To accomplish this, runtime security focuses on:

- Observing live workload behavior instead of static configurations
- Identifying unexpected or malicious activity, such as unauthorized processes or unusual network connections
- Providing continuous visibility across hosts, containers, serverless, and cloud environments

This visibility is especially important in cloud-native environments where workloads are constantly changing. Containers spin up and down, APIs communicate across distributed services, and attackers increasingly operate using legitimate tools and stolen credentials. Runtime security helps teams detect this activity with the context needed to investigate and respond effectively.

How Runtime Security Works

Collecting Runtime Telemetry

Runtime security begins with continuously capturing system activity across workloads, including:

- Process execution and command-line activity
- File access and system changes
- Network connections and traffic patterns

This telemetry is gathered through lightweight agents using kernel-level technologies such as eBPF, enabling deep visibility with minimal performance impact.

The quality of runtime telemetry directly affects how effectively threats can be detected and investigated. Shallow visibility may identify isolated events, but deeper telemetry provides the context needed to understand how activity relates across workloads, users, processes, and cloud infrastructure.

This becomes especially important when investigating sophisticated attacks that rely on legitimate processes, credential misuse, or zero-day vulnerabilities. The more detailed the telemetry, the more accurately security teams can distinguish normal behavior from malicious activity.

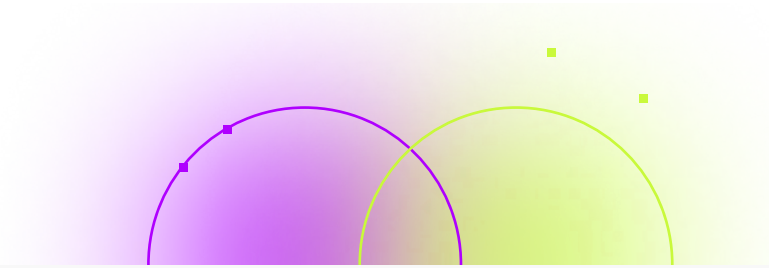
Monitoring Workload Behavior

Once telemetry is collected, runtime security analyzes workload behavior over time by:

- Establishing baselines of normal activity
- Tracking deviations from expected behavior
- Identifying patterns that indicate misuse or compromise

This allows security teams to move beyond static rules and understand how workloads actually behave in production environments.

For example, runtime security can identify when a container begins making unexpected outbound connections, when a process launches outside its normal execution pattern, or when privileged activity suddenly changes. These behavioral signals help surface threats that traditional preventive controls may miss after deployment.



Detecting and Correlating Threats

Runtime signals are analyzed and correlated to surface meaningful threats, such as:

- Suspicious process execution within containers
- Unexpected outbound connections
- Lateral movement across workloads

Effective runtime security connects these signals with additional context like identity, configuration, workload history, and cloud activity to improve accuracy and reduce noise.

Rather than generating isolated alerts, runtime security helps teams understand how events relate to one another. This context is critical for identifying attack chains, prioritizing risk, and reducing false positives in dynamic cloud environments.

Responding and Investigating Threats

When suspicious activity is identified, runtime security enables:

- Alerts with supporting evidence and context
- Investigation into process lineage and activity history
- Manual or automated response actions

The goal is not just to generate alerts, but to help security teams quickly understand what happened and determine the appropriate next step.

Modern runtime security platforms can support actions such as killing malicious processes, quarantining affected hosts, or tracing suspicious activity back to its source. Faster investigation and response help reduce attacker dwell time and limit the impact of active threats.

Why Runtime Security Matters in Modern Cloud Environments

Modern cloud environments introduce challenges that make runtime visibility essential:

Ephemeral infrastructure
Containers and workloads are constantly created and destroyed, making point-in-time visibility insufficient

Expanding attack surface
Microservices, APIs, and distributed architectures increase opportunities for exploitation

Limits of preventive controls
Vulnerability scans and posture checks cannot stop attacks that occur after deployment

Real attacker behavior
Threat actors operate during execution, often using stolen credentials and legitimate tools to avoid detection

Zero-day vulnerabilities
Runtime security helps detect malicious activity when attackers exploit vulnerabilities that are not yet known or patched.

Runtime security provides visibility at the point where attacks actually occur.

Key Capabilities of Runtime Security

Runtime security solutions provide the core capabilities organizations need to detect and respond to threats during execution.

Continuous workload visibility
Runtime security provides insight into processes, network activity, system call and system behavior across cloud workloads, containers, and hosts. This visibility helps teams understand what is happening inside active environments rather than relying on periodic snapshots.

Behavior-based threat detection

Instead of relying only on known signatures or static rules, runtime security identifies suspicious activity based on deviations from expected behavior. This improves the ability to detect credential misuse, malicious processes, and emerging threats.

Broad workload coverage

Modern environments often span containers, virtual machines, Kubernetes clusters, serverless infrastructure, and hybrid cloud environments. Runtime security helps provide consistent monitoring across these distributed systems.

Context-rich alerting

Alerts are enriched with details such as process lineage, identity data, workload history, and cloud context. This additional visibility helps teams investigate incidents faster and reduce alert fatigue.

Investigation and response support

Runtime security platforms support investigation workflows and response actions that help teams contain threats, understand root cause, and improve remediation efforts.

Runtime Security and Compliance Requirements

Runtime security can also help organizations support compliance initiatives by improving visibility, monitoring, and incident response capabilities across cloud environments.

Many regulatory frameworks require organizations to maintain continuous monitoring, detect suspicious activity, and protect sensitive systems from unauthorized access. Runtime security supports these efforts by providing real-time visibility into workload behavior and helping security teams investigate threats more effectively.

Examples include:

- **PCI DSS 4.0**, which emphasizes continuous security monitoring and detection capabilities
- **SOC 2**, which requires organizations to maintain controls around security, availability, and incident response
- **HIPAA**, which requires safeguards to protect sensitive healthcare data and detect unauthorized activity

While runtime security alone does not guarantee compliance, it helps organizations strengthen operational security controls that support broader compliance programs.

Runtime Security in the Cloud Security Stack

Runtime security works alongside other cloud security controls, but each technology addresses different stages of risk detection and remediation.

Capability	Runtime Security	CSPM	Vulnerability Scanning
Primary Focus	Active threat detection and response	Cloud misconfiguration detection	Known vulnerability identification
Timing	Runtime / post-deployment	Pre-deployment and configuration review	Pre-deployment and periodic scanning
Visibility	Live workload behavior and activity	Cloud resource configurations	Packages, libraries, and software versions
Detects Credential Misuse	Yes	No	No
Detects Active Threats	Yes	Limited	No
Supports Runtime Response	Yes	No	No
Typical Use Cases	Threat detection, investigation, and response	Compliance and posture management	Vulnerability remediation
Best Used With	CSPM, CIEM, vulnerability scanning	Runtime security and CIEM	Runtime security and posture management

Runtime security does not replace posture management, vulnerability scanning, or identity controls. It complements them by providing visibility into active environments and detecting threats during execution.

- CSPM (Cloud Security Posture Management) identifies misconfigurations before deployment
- CIEM (Cloud Infrastructure Entitlement Management) manages identity and access risks
- Vulnerability scanning detects known weaknesses in images and code

Runtime security complements these by:

- Observing how workloads behave after deployment
- Detecting threats that bypass preventive controls
- Providing visibility into active environments
- Protecting from malicious activity by killing malicious processes and quarantining hosts.

It acts as the bridge between posture, identity, and real-world activity, helping teams understand risk in context.

Runtime security becomes more powerful when detections can be traced back to code and configuration changes. By tying runtime detections directly to the line of code that introduced the behavior, security teams can move beyond identifying issues to understanding their origin. This creates a code-to-cloud security model where risks identified at runtime can be traced, prioritized, and fixed at the source.

How Uptycs Delivers Runtime Security

Runtime security is only as effective as the telemetry behind it.

Uptycs takes a telemetry-first approach, capturing detailed system activity and normalizing it into a unified data model. This allows security teams to correlate signals across workloads, identities, configurations, and cloud logs in a single platform.

By connecting runtime detections back to the originating code and configuration, Uptycs helps teams close the gap between development and production. This improves root cause analysis, accelerates remediation, and supports a broader code-to-cloud security model.

Because Uptycs uses a lightweight sensor, organizations can deploy visibility across workloads without introducing significant operational overhead. This makes continuous monitoring practical across cloud-native environments and enables deep, in-memory visibility into runtime activity.

This combination of telemetry depth, broad workload coverage, and contextual correlation helps security teams investigate threats more efficiently and reduce operational complexity.

With Uptycs, teams can:

- Gain deep visibility into Linux workloads using eBPF
- Correlate runtime activity with posture, identity data, and cloud logs
- Investigate alerts with full context without switching tools
- Focus on high-fidelity signals to reduce noise
- Leverage high-fidelity Linux EDR telemetry to improve runtime visibility, detection depth, and investigation context

This approach helps organizations move from isolated alerts to a clearer understanding of risk across their environments, improving both detection quality and investigation efficiency.

Explore Related Resources

- **From Code to Container: Building an End-to-End Security Pipeline with Policy-Based Admission**
  Explores code-to-cloud security and runtime detection using eBPF.
- **Application Posture Management: Securing the Code-to-Cloud Journey**
  Examines how runtime security complements posture management across cloud environments.
- **Introducing Uptycs EKS Add-ons: Advanced Kubernetes Security for AWS Users**
  Highlights runtime visibility and detection for Kubernetes workloads.
- **Breaking Down the Silos in Kubernetes Security with Uptycs Unified Risk Insights**
  Shows how runtime findings can be correlated with posture and risk context.
- **Uptycs Platform Overview (CNAPP)**
  Overview of unified telemetry-driven detection and response from code to cloud.

Frequently Asked Questions

1. What is runtime security in cloud environments?
Runtime security is the practice of monitoring and protecting workloads while they are actively running. It focuses on detecting threats based on real behavior rather than static configurations or pre-deployment scans.

2. How is runtime security different from vulnerability scanning?
Vulnerability scanning identifies known weaknesses before deployment, while runtime security detects threats during execution. Runtime security focuses on how workloads behave, not just what vulnerabilities they contain.

3. Why is runtime security important for containers and Kubernetes?
Containers and Kubernetes environments are highly dynamic, with workloads constantly starting and stopping. Runtime security provides continuous visibility into this activity, helping detect threats that occur after deployment.

4. What types of threats can runtime security detect?

Runtime security can detect:

- Unauthorized process execution
- Suspicious network connections
- Lateral movement across workloads
- Misuse of legitimate tools within the environment
- Use of stolen credentials

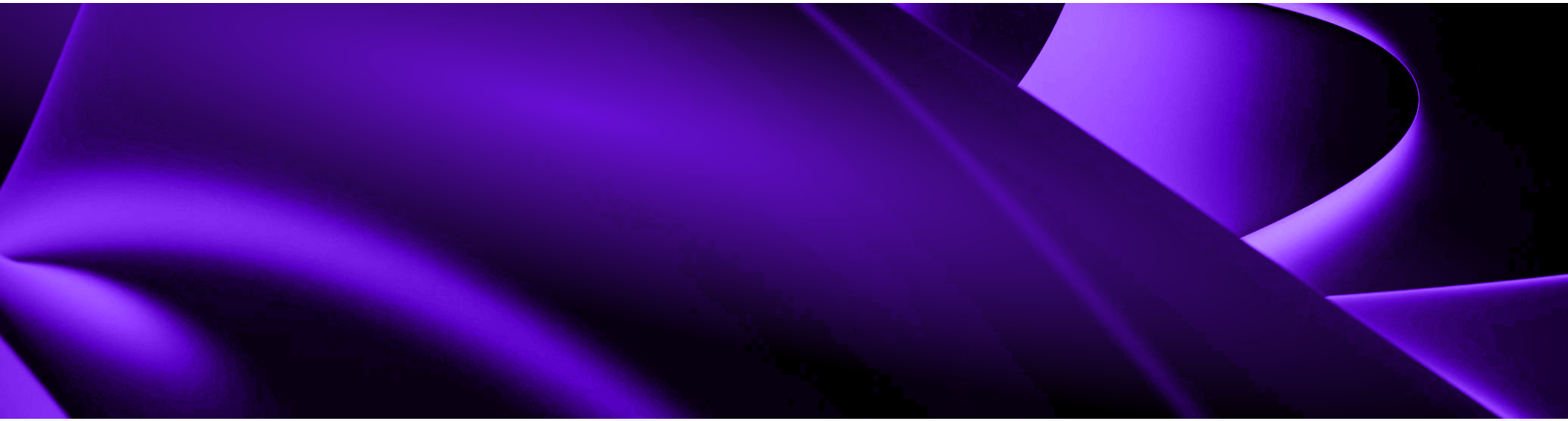
5. What is runtime telemetry and why does it matter?
Runtime telemetry refers to the data collected from workloads during execution, including processes, network activity, and system changes. It provides the foundation for detecting and investigating threats in context.

6. How does runtime security reduce false positives?
By analyzing behavior over time and correlating signals with context such as identity and configuration, runtime security improves accuracy and reduces alert noise compared to static rule-based tools.

7. Where does runtime security fit in a CNAPP strategy?
Runtime security complements posture management, identity controls, and vulnerability scanning by providing visibility into active workloads. It helps connect these signals to show risk in context.

8. How does Uptycs approach runtime security?
Uptycs takes a telemetry-first approach to runtime security, capturing detailed system activity and correlating it across workloads, identities, and configurations. With a lightweight sensor and deep runtime visibility, Uptycs helps teams detect threats more accurately, investigate faster, and connect detections back to the code and configuration behind them.

9. How does runtime security improve my vulnerability management posture?
Traditional vulnerability scanners flag everything that could be vulnerable based on package versions — leaving security teams buried in noise. Runtime security cuts through that by revealing what's actually exploitable right now: is the vulnerable package loaded by a running process, is it exposed to the internet, and is it running with elevated privileges? This context powers risk-based prioritization that fuses runtime signals with CVSS scores, EPSS exploitability data, and asset criticality — so your team remediates what matters most, not just what scores highest on paper. The result is a tighter, evidence-backed remediation queue that shrinks real exposure faster.



uptycs

Uptycs delivers the industry's first **Unified CNAPP** with integrated XDR, powered by Juno - the only verifiable AI Security Analyst.

Unlike traditional CNAPPs that rely on static snapshots, Uptycs unifies cloud configuration data with deep runtime telemetry across cloud, containers, and endpoints into a single Unified Ontology. Trusted by the Fortune 500, Uptycs powers Continuous Threat Exposure Management (CTEM) by normalizing telemetry into a single data lake, enabling the Juno AI Analyst to provide **"Glass Box" transparency**, turning hours of investigation into minutes of verified answers.

Learn more at uptycs.com/juno-ai