SPYDERBAT

Spyderbat Guardian

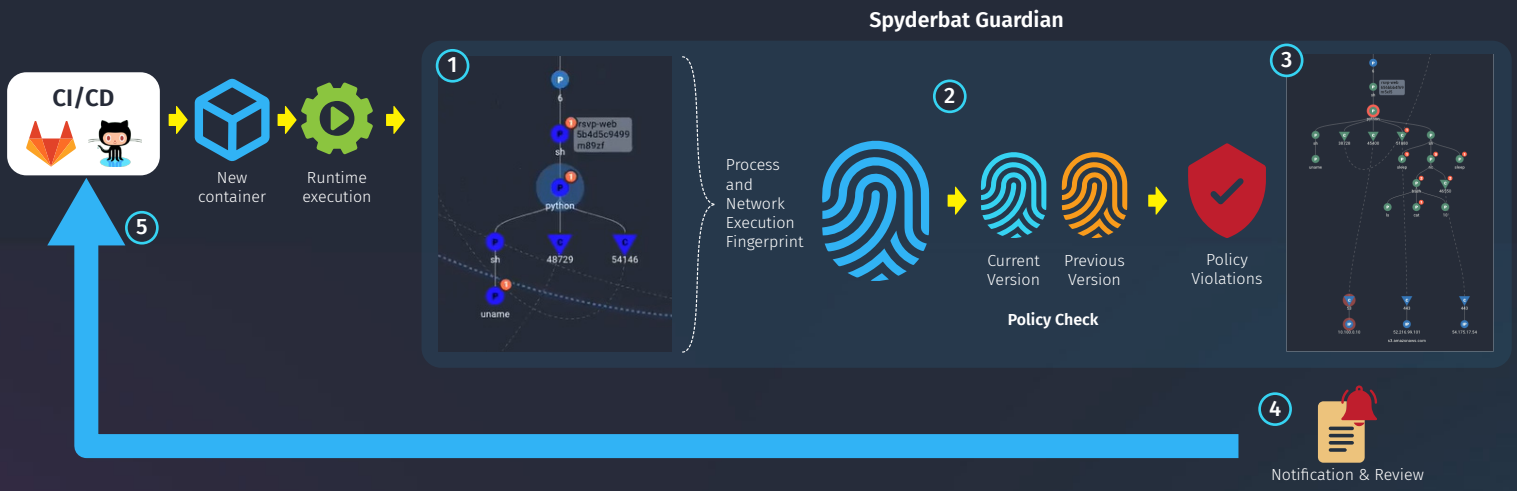
Automatically protect known-good application behavior to eliminate drift and correct anomalies

Overview

Containerized applications are composed of hundreds, sometimes thousands, of discrete components, all of which are interconnected and orchestrated via desired state policy. Automation and fast iteration continuously updates your components leading to unintended drift and errors, causing downtime and security risk. Spyderbat Guardian automatically identifies your known-good application workload behaviors, immediately detecting when the intended state drifts from the observed state. Without wasting cycles digging through logs, your developers, platform teams, and security teams use Guardian to quickly remediate unintended drift.

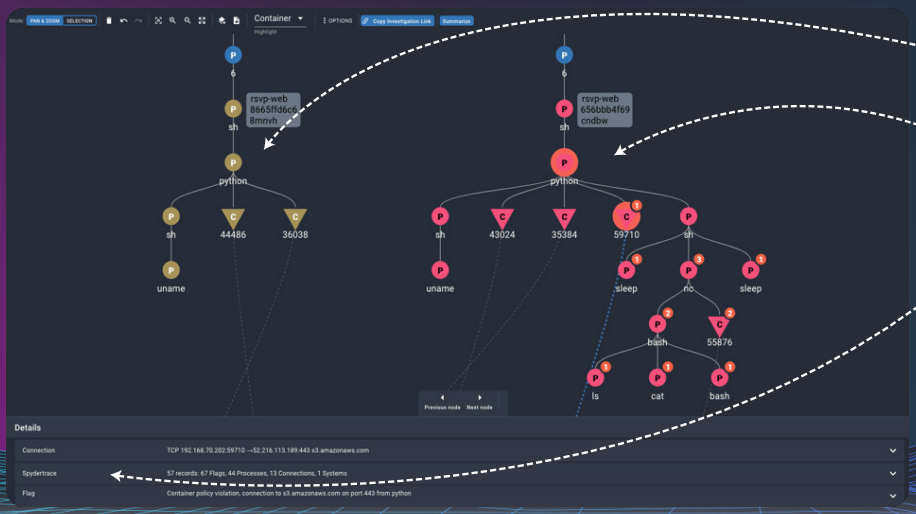
Challenge	Guardian Benefits
Understand your workload behaviors	Captures your workload behaviors' process lineage, including environmental variables, user rights, and network activity, with immediate insight to root causes.
Remove risks to uptime performance	Secures your validated workload behaviors, auto-correcting drift in production while preserving post-incident visibility.
GitOps integration	Programmatically validates workload behavior fingerprints throughout the software development lifecycle.

Technical Details



Guardian continuously analyzes and fingerprints workload behaviors, then compares them to previous versions to instantly recognize drift.

1. As developers build new code, Guardian automatically fingerprints the workload behavior of new containers, capturing the process execution, user permissions, network connections, and other details.
2. Guardian automatically compares the new container profile to the previous version or to a validated “known-good” version.
3. The root cause of drift is captured even if automated actions restart the pod/container back to known-good states.
4. Guardian outputs the workload behavior in YAML, enabling programmatic integration with Policy-As-Code and GitOps processes.
5. Through the CI/CD process, developers validate new the workload behaviors that are required for new app functionality, while also catching any unintended resource uses, misconfigurations, or even supply chain compromises before reaching production.



1. Previous workload fingerprint including process and network execution.
2. New deviations are ‘flagged’ as policy violations - but do not result in alerts.
3. Spyderbat instead tracks the full series of activities as “Spydertraces,”, scoring the entire trace for prioritization and automation

Customer Case Study

The Situation: A third-party component in a staging environment was misconfigured with values used in development environments, causing incorrect network traffic.

The Impact: The build failed system tests, but root cause was unclear. Manual investigation delayed the release and distracted multiple engineers from new feature development.

The Resolution: Spyderbat Guardian automatically recognized the workload runtime behavior differed from previous versions, immediately notifying the development team via Slack leading to a quick correction of the misconfiguration.

To see Guardian’s fingerprinting in action and other key capabilities of the Spyderbat platform, **start your free trial today:** www.Spyderbat.com