sysdig

WHITE PAPER

Runtime Insights are Key to Shift-Left Security

Cloud security programs often emphasize one of two approaches: shift left or shield right. Shift-left approaches focus on processes and tooling that promote secure design and pre-release testing to identify security issues before they become production problems. Shift-left approaches are heavily intertwined with DevOps practices and aim to prevent breaches through hardening security posture. Shield-right approaches focus on operational practices, security monitoring, and runtime security mechanisms to prevent security incidents, as well as detect and respond to events as they occur. Both approaches are essential to a mature cybersecurity program, but in practice, these approaches often run in isolation which leads to silos in the organization.

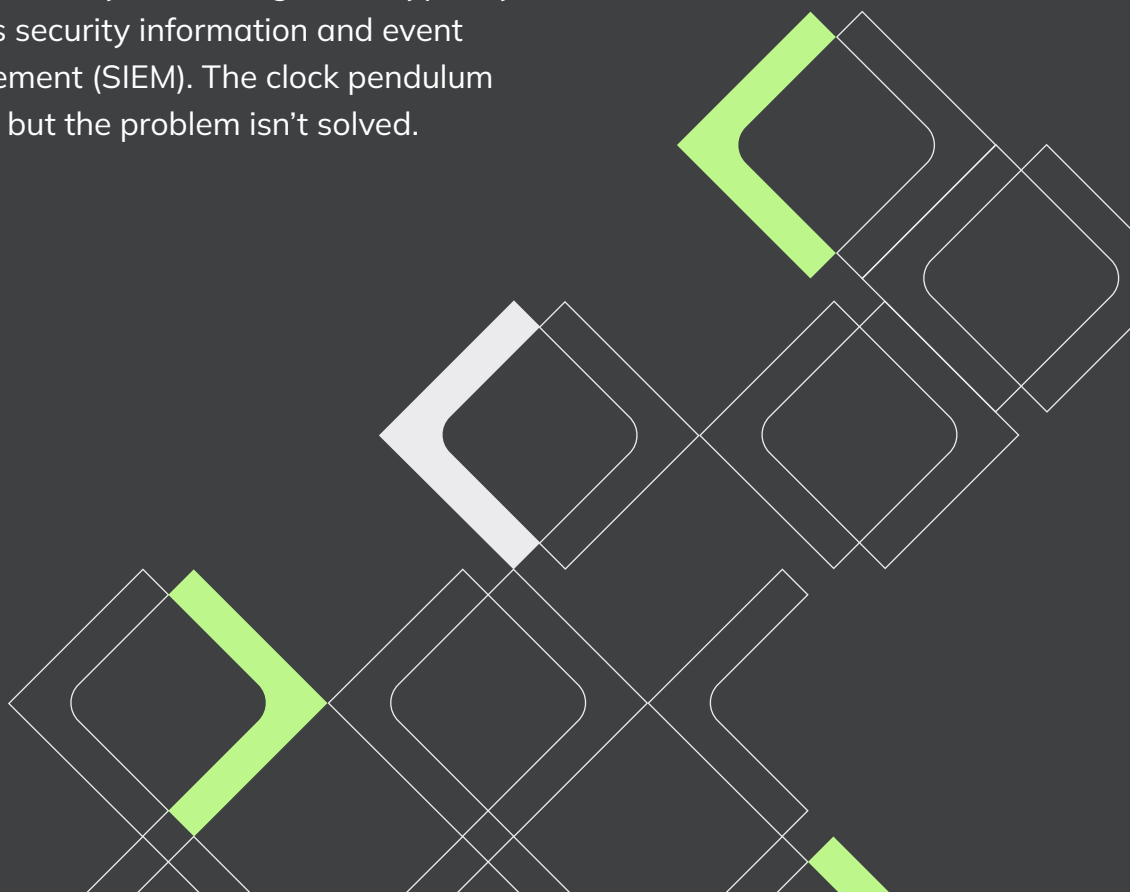
Runtime insights are the glue between these two worlds, empowering organizations to keep pace with the speed and sophistication of cloud attacks and scale their cybersecurity. Applying a security approach that incorporates runtime insights enables organizations to prioritize and mitigate risk, detect and respond to threats in real time, and identify risky combinations across environments. This paper explores the importance of runtime insights for shift-left activities or preventative security, helping you avoid attacks on your organization's innovation in the cloud.

Table of Contents

The challenge with current shift-left strategies

Shift-left security has a noise problem. Organizations start their journey with pre-release scanning tools, and quickly drown in a deluge of scanner output. It's a struggle to find an efficient way to pass or fail application releases. Finding vulnerabilities is never a problem, but determining if a risk must be addressed is challenging.

Given the flood of security issues with no clear risk prioritization, organizations subsequently decide to refocus attention to runtime security. They'll often start with outdated blocking mechanisms, like next-generation firewalls or web application firewalls, and they'll amp up their security monitoring which typically includes security information and event management (SIEM). The clock pendulum swings, but the problem isn't solved.

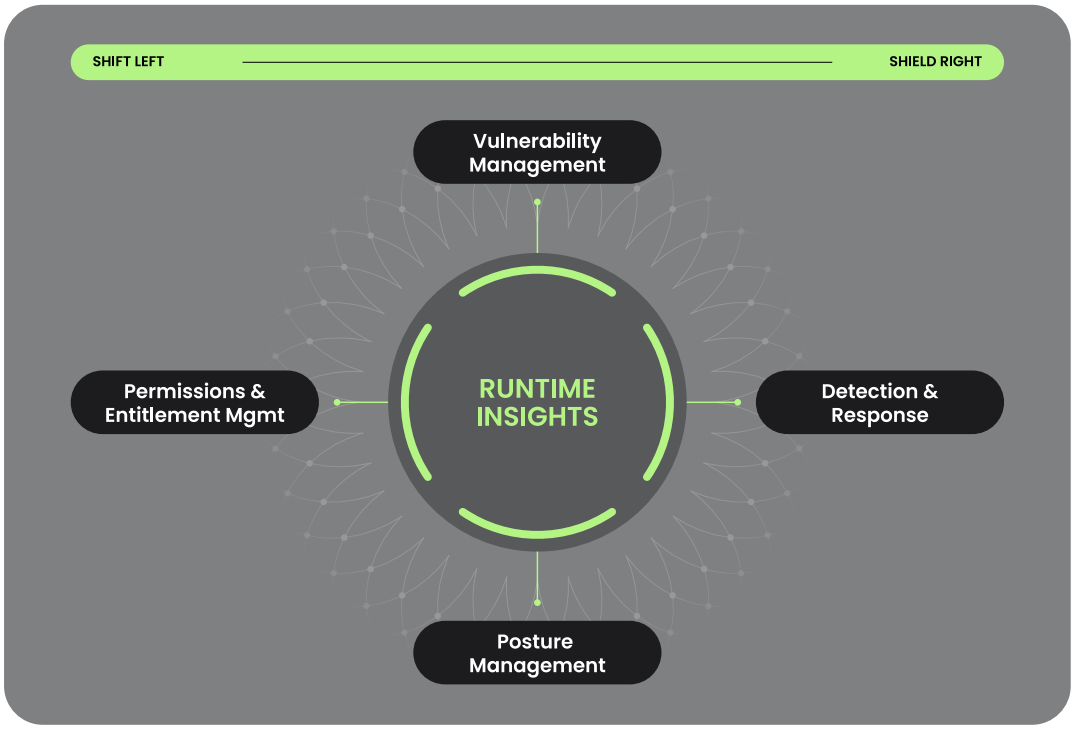


Cloud transformation creates security gaps

As with any major change in business practices, team restructuring and process refinements are a key part of cloud adoption. Organizations evolve to include DevOps, Cloud Engineering, and Platform Operations functions. Organizations may also take a different course where they form security centers of excellence that oversee cloud security processes and govern environments, or they embed individuals within engineering teams.

Organizations also look to tools to address gaps in their cloud security strategy. For prevention, organizations often focus on point-scanning tools for posture management, vulnerability management, and permissions and entitlement management. For defense, organizations often focus on security monitoring including traditional tools like SIEM. But this isn't entirely a proactive or protective approach, and it leans more into threat detection and response.

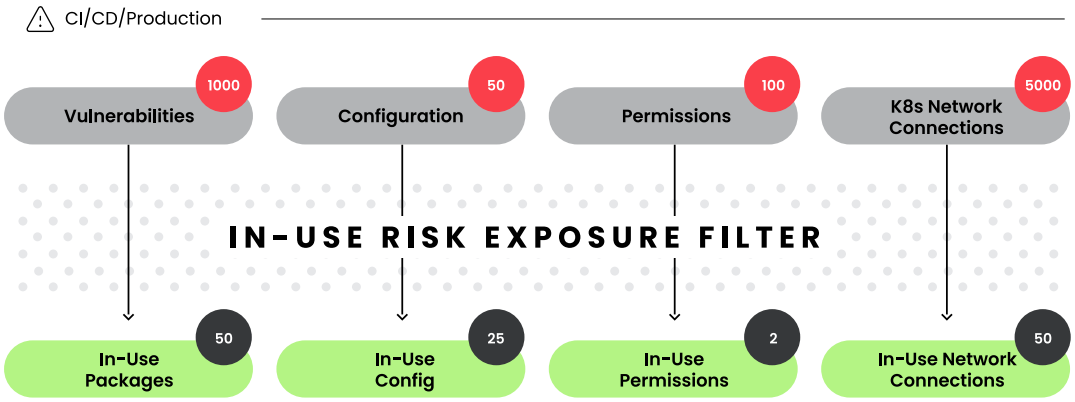
The emergence of runtime insights has paved the way for solutions that consolidate tools for these domains, providing real-time information about what's in-use so security teams can prevent and defend with greater confidence. The following image depicts where these security activities usually land on the shift-left and shield-right spectrum.



Applying runtime insights to security

Runtime insights provide actionable information to prioritize the most impactful problems in your environments based on the knowledge of what is running right now. They provide a lens of what's actually happening in deployments, allowing security and development teams to address what matters most. We're not talking about the old language of firewalls or IPS. We're referring to capabilities that increase visibility for deployed applications and systems as opposed to relying solely on pre-delivery scanning, where most secure design and security testing approaches fall short.

The figure below visualizes how the number of detected environmental problems increases with shift-left scanning, and how the use of an appropriate exposure filter can reduce the number of problems into something more actionable based on actual risk.



Key attributes of runtime insights:

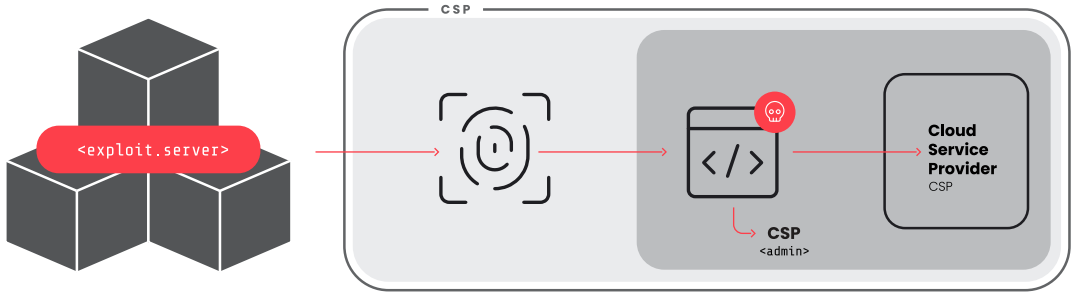
In order to evaluate and secure all types of environments and workloads, runtime instrumentation becomes essential. The four primary characteristics and values of runtime insights are as follows:

- **In-Use:** Aid teams in prioritizing risks so they can filter out noise and scale their security program effectively. Highlight what's actually used by the organization and its systems to help reduce the burden of chasing false positives and less severe issues.
- **Multi-Domain Correlation:** Identify risky combinations across environments that create attack paths to sensitive data. Enhance data visualizations that highlight where gaps are present in environment configuration as a type of preventive control.
- **Real-Time Detection:** Enable continuous and real-time detections as cloud environments are ever changing. Point-in-time environmental assessments and scans don't cut it. Tolerance for latency in data collection or analysis is also extremely low since it creates massive windows of exposure.
- **End-to-End Detection:** Detect on everything in cloud environments composed of servers, containers, cloud services, and serverless functions by using a variety of approaches, including rules, behavior-based detections, and ML-based detections. This attribute is covered in the paper Securing the Cloud with End-to-End Detection.

Examining an attack chain where runtime insights help

Using runtime insights to inform shift-left and shield-right activities, we're able to prevent, detect, and remediate issues that are exploited in complex attack chains in threats such as SCARLETEEL. This attack chain, discovered by the Sysdig Threat Research Team and depicted in the figure below, was more sophisticated than most. It started from a compromised Kubernetes container and spread to the victim's AWS account. The attack chain has many facets, which spotlights the inherent complexity of securing cloud-based infrastructure. Pre-delivery scanning and vulnerability management alone would not mitigate all the risk in this scenario. Runtime insights help accelerate detection, enhance visibility, and spotlight the attack path.

Surfacing what permissions are actually in-use enables us to optimize certain roles that may be over-privileged (such as ability to write to an S3 bucket) or that can result in unwanted infrastructure configuration drift (such as permission to deploy new code directly in production, bypassing build pipelines). Also, advising on what packages are deployed and in-use with known vulnerabilities helps focus team efforts on prioritizing remediation to prevent a possible incident or breach.



Three ways that runtime insights boost security

Let's take a deeper dive into how runtime insights and these key attributes apply to the three shift-left security activities described earlier: vulnerability management, posture management, and permissions and entitlement management.

1. Vulnerability management

Security and engineering teams are fatigued by endless lists of vulnerabilities to sort through. As of May 2023, there are over 10,000 new vulnerabilities registered in the CVE database, adding to the existing pool of over 210,000 vulnerabilities. In a cloud environment where a heavy shift-left approach is used, vulnerabilities are detected by scanners at different stages of design, code commit, build, and delivery; therefore, similar issues may be found repeatedly, which complicates release decisions. Organizations need accurate and timely inventory of all affected assets so they can prioritize fixes or dependency changes before attackers are able to exploit them. Unfortunately, these requirements are at odds with fast release cadences and accelerating release velocity.

Successful, modern vulnerability management requires that security teams prioritize vulnerabilities based on the actual or real risk to their organization. Vulnerability prioritization is essential to reducing the fatigue of engineering teams, ensuring iterative application development is safe and high quality, and maintaining fast release cadences. Risk-based decisions are foundational to all security programs, and relevant criteria for vulnerability management include:

- What vulnerabilities are exploitable?
- What vulnerabilities have known exploits or proof-of-concept code available?
- What's being actively targeted in the wild or in specific industries?
- Where are vulnerabilities present in all environments and in all dependencies?

Sysdig's findings in the Sysdig 2024 Cloud-Native Security and Container Usage Report provide signs of hope for overburdened developers by focusing remediation efforts on vulnerable packages loaded at runtime. While the vast majority of images include a high or critical severity vulnerability, the percentage of these that present real risk at runtime is much lower. Consider the entire landscape of vulnerabilities seen in customer deployments, visualized in the subsequent figure. Of workloads with critical or high severity vulnerabilities:

- 86% have a fix available
- Only 8% are fixable and actually present in runtime
- Only 1.2% are fixable, in-use at runtime, and have known exploit code

By filtering on what's actually fixable, in-use, and exploitable, we are better able to prioritize any mitigation or remediation. Vulnerabilities of higher risk can be prioritized, relative to the organization's unique environment and application design choices. These vulnerabilities are likely the ones that expose the organization to real and urgent danger.

When exploitable vulnerabilities must remain in your environment, security teams can reduce the risk of compromise by implementing runtime security detections. Runtime protection is often powered by rules, but it should also employ a multi-layered approach that incorporates behavior anomaly detection and AI- or ML-based detection. This approach improves detection and mitigation of zero-day exploits and unknown threats. Runtime protection mechanisms can also be tuned to detect novel threats that target vulnerable workloads within the unique environments of organizations.

By filtering on what's actually fixable, in-use, and exploitable, we are better able to prioritize any mitigation or remediation. Vulnerabilities of higher risk can be prioritized, relative to the organization's unique environment and application design choices.

2. Posture management

Hardening infrastructure that powers applications and services is fundamental to all security program work. Unneeded cloud services and known vulnerable configurations should be disabled from the outset, ideally using infrastructure-as-code and policy-as-code approaches. And configurations should be continuously validated to ensure an old misconfiguration doesn't creep back into the environment. Evaluating security posture is a tricky proposition. Organizations often try a variety of approaches, including the native cloud provider management consoles and auditing mechanisms, shell scripts for auditing, and open source tools like OpenSCAP.

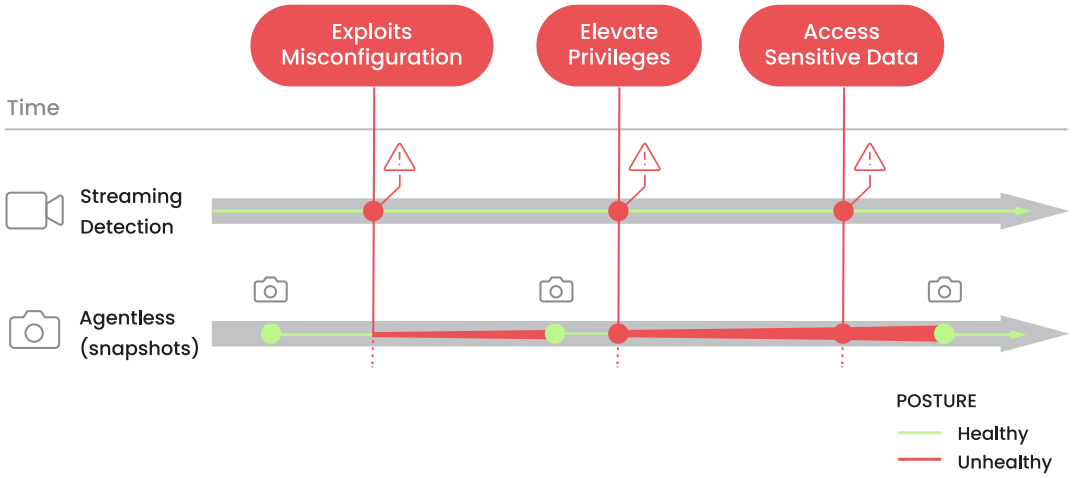
For organizations further along in cloud adoption, it's common to see deployment of a cloud security posture management (CSPM) tool to audit and report on misconfigurations in the organization's tenants based on snapshots of the environment. A CSPM tool also validates controls and determines if configurations follow security guidelines, best practices, and standards. Minimums for security controls are defined in standards such as CIS benchmarks, DISA STIGs, and NIST SPs, regulatory requirements like HIPAA, or industry-specific standards like PCI DSS.

Governance and compliance headaches are commonplace, even in spite of CSPM deployments. GRC teams define standard security requirements. Engineering teams are expected to satisfy these requirements as they deploy new applications and systems, and ideally it's all stored as "golden images" in sanctioned repositories or registries. However, two common pitfalls arise:

1. Gaps between broader security requirements (useful for compliance or legal purposes) and specific technical implementation details to meet those requirements.
2. Deviations from the planned, secure configuration and what's delivered or maintained over time, as environmental changes occur normally throughout build, delivery, and operation.

Many organizations are using point-in-time assessments, even in cases where a CSPM tool is deployed. In industry, these types of validations are often referred to as snapshot approaches. Similar to a penetration test, it's best to view these types of validations as useful for satisfying compliance audits or regulatory requirements that mandate checks at regular cadence. However, such approaches aren't a high bar for security. Snapshot approaches are high latency and low accuracy.

This minimal form of security provides a false sense of assurance. Running a series of scheduled audits against static controls cannot assure you that configurations will remain unchanged in the near future. Misconfigurations are windows of exposure for attackers that are regularly targeted and exploited. Having a live inventory of your cloud assets and corresponding security posture is the best way to keep your organization safe from unwanted configuration changes. The image below visualizes streaming detection and snapshot approaches, spotlighting the resulting window of exposure from an agentless-only approach.



The U.S. National Security Agency, for instance, has written that cloud misconfigurations are "the most prevalent cloud vulnerability" and "security in the cloud is a constant process and customers should continually monitor their cloud resources and work to improve their security posture." [1]

The scalability and extensibility of cloud services drive innovation, but attackers are leveraging this expanded attack surface to move laterally across environments after gaining initial access. When combined with other risks, cloud misconfigurations can create hidden attack paths that attackers can exploit to access sensitive information. With runtime insights, organizations can visualize these hidden paths by identifying combinations of risks across domains in their cloud infrastructure. By seeing which areas of their security posture are at risk at runtime, they can identify gaps in environment configuration and prevent them from being exploited.

In a fast-changing cloud-native system, application owners need to make changes to their applications and underlying infrastructure to continuously adapt the product to satisfy new business requirements. As these modifications occur, it's also necessary to adjust the configuration of applications and infrastructure, which can bring systems out of the state of fortified posture. This is what we define as configuration or posture drift.

The challenge organizations face with posture drift is how to detect it in real time, catching it before there are exploitable conditions that can lead to an incident or breach. To adequately support "as-code" approaches, detection should also provide contextualized remediation and generate pull requests to avoid further wasted time. Being more flexible in the response, leaning into risk prioritization to reduce noise, embracing automation with "as-code" approaches, and enriching alerts on misconfigurations with runtime insights alleviates many of the headaches inherent with posture assessment and enforcement.

1 The U.S. National Security Agency, Mitigating Cloud Vulnerabilities, 22 January 2020

3. Permissions and Entitlement Management

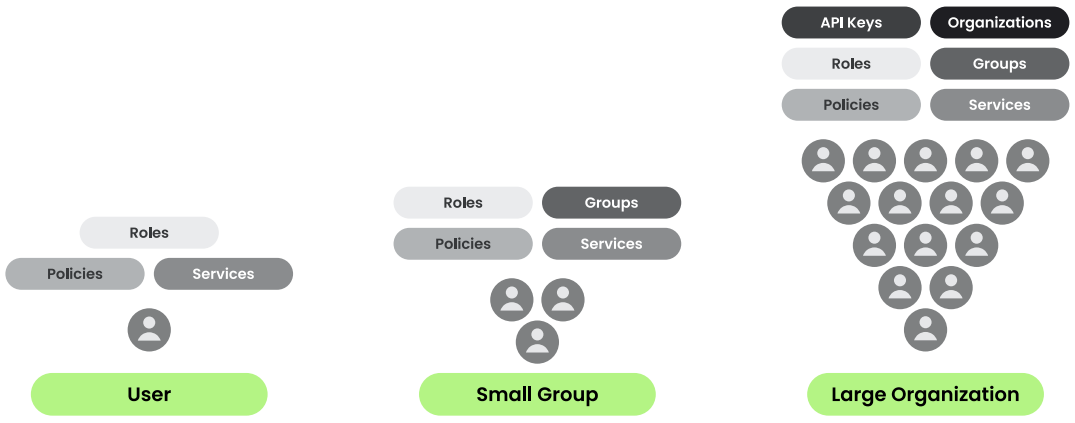
Misconfigurations are still the biggest player in security incidents and, therefore, should be one of the greatest causes for concern in organizations. Many misconfigurations are related to permissions or access controls rather than traditional infrastructure configuration.

According to Gartner®, "by 2023, 75% of security failures will result from inadequate management of identities, access, and privileges, up from 50% in 2020." [2] Although many organizations are talking about zero-trust principles, such as enforcing least privilege, our data shows little evidence of action with 98% of cloud permissions going unused. [3]

We learn about the importance of access control repeatedly as security practitioners, yet identity and access control missteps remain highly prevalent. Why does this keep happening? In practice, systems use a mashup of different access control types depending on the complete architecture, including discretionary access control (DAC) and role-based access control (RBAC).

How you assign identities to groups or roles to grant permissions and privileges varies based on the access control type and technology stack. Users are part of many groups. Users and groups are mapped to many roles. Roles are overly-broad so they are usable at the expense of granularity and tight access control. Cloud resources are numerous, so organizations will keep access control coarse-grained so as not to introduce more complexity or fragility. Permissions also change over time as a result of functionality changes, employee turnover, employee job changes, customer attrition, changes to technology stacks, and more.

The end result is partial chaos, regardless of your identity and access management (IAM) team's structures and processes. Pockets of identity and permissions form rapidly, as seen in the figure below. However, these islands of distributed access still need to be connected and integrated as part of modern design. Enforcing strict access control quickly becomes untenable for most organizations.



2 GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved. Gartner, Best Practices for Optimizing IGA Access Certification, Gautham Mudra, 4 April 2022

3 Sysdig, 2024 Cloud-Native Security and Usage Report, January 2024

The least privilege principle is absolutely critical to access controls. Any developer, security architect, or compliance expert should be able to do their work without blockers, but they should also be unable to go beyond this scope. And we must not forget the machine identities or service accounts that have their own authorizations that must also follow least privilege. In cloud environments, machine identities can often outnumber human identities by virtue of abstractions, integration, and automation.

Least privilege by itself is no longer enough for robust access control. We must continuously validate authorization and always assume that a given environment is compromised. Organizations want to continuously monitor identity, application, and system behavior to detect and enforce on suspicious activity, effectively zero trust architecture (ZTA). A given threat actor could have sufficient privileges to cause damage if they manage to take over other accounts or harvest secrets. And the damage can be done without elevating privileges or abusing misconfigurations.

Effective IAM requires collaboration and ownership among many different teams to maintain granularity in the assignment of permissions to accounts and non-human users. Each team should know its scope of responsibility and minimum resources needed. IT teams, or IAM teams if they exist, should follow the principle of least privilege using the controls that cloud providers offer. Unfortunately, CloudOps and PlatformOps teams, if they are present, are operating on requirements set forth by teams that may be disconnected from the operation of environments. Disparity between mandated security requirements and specific technical implementations quickly rears its head.

Teams charged with access control often struggle with cloud permissions and the mix of human and machine identity. Permissioning is likely also delegated to data owners and custodians in scalable data security approaches. Determining what access is needed, when it's needed, and if it's actually in-use is a guessing game. The only way to get ahead of this dilemma is to understand what identity has been provisioned and what access is granted, and then pair this with real-time access patterns (or in-use permissions) to accurately model permissions based on observed behavior. Common types of permissions missteps that can be surfaced with this type of approach include unused permissions, unused administrative accounts, privileged accounts lacking 2FA, excessive permissions, and privileged users doing something destructive, like tearing down a production cluster.

Least privilege by itself is no longer enough for robust access control. We must continuously validate authorization and always assume that a given environment is compromised.

Runtime insights keep your applications secure

Clearly, security gaps occur in the cloud, and cloud security strategy needs thorough examination. Tooling that's currently in place may not be suitable for all cloud environments, particularly when environments are cloud-native. We've identified some of the core traits that can address these gaps, particularly runtime instrumentation, risk prioritization, and real-time detection. Organizations should look to unified capabilities or platforms to avoid some of the disconnects that occur with point solutions. This is all the more critical as shift-left or shield-right priorities take hold. Any tooling or platform at a minimum should:

- Enrich with the right environment and workload signals
- Provide and prefer application and service context over host context
- Ingest relevant event sources and process in place
- Use instrumentation as appropriate for different workload types
- Correlate pre-delivery scan results with runtime monitoring
- Integrate with respective team workflows and systems to avoid disruption
- Tailor and contextualize remediation for the organization's unique deployments
- Facilitate automation with "as-code" and API-first approaches

These capabilities were historically delivered through other tooling categories, including CSPM, cloud infrastructure entitlement management (CIEM), and cloud workload protection (CWP). Because of the intersections that form naturally in how such tools get deployed and operated, a new combined platform approach emerged in the form of cloud-native application protection platforms (CNAPP).

As cloud environments become more complex, the value of a platform that can correlate context and findings across domains within cloud infrastructure becomes clear. Runtime visibility enables organizations to identify and prioritize the most important risks across servers, containers, cloud services, serverless functions, and identities through multi-domain correlation. By leveraging the power of runtime insights, security teams can visualize potential attack paths and surface the context they need to harden their security posture and prevent attacks before they happen. Regardless of which tooling category you seek to augment your cloud security program, ensure that any tool is equipped with runtime insights and provides end-to-end detection in order to keep your cloud environments safe.

Recommended reading

- Run Faster, Runtime Followers
- Why Companies Still Struggle with Least Privilege in the Cloud
- Vulnerability Prioritization – Combating Developer Fatigue
- Real-Time Threat Detection in the Cloud
- SCARLETEEL: Operation leveraging Terraform, Kubernetes, and AWS for data theft
- Will the Cloud End the Endpoint?
- Sysdig Learn Cloud Native
- SANS CNAPP Buyers Guide

WP-006 REV. B 02/24