The Cloud SecOps Paradox

Why Security Operations Teams Escalate Too Much and Catch Too Little in the Cloud

Stream.Security

Table Of Contents

Introduction

You're responsible for securing the cloud. But you don't own it.

If you're working in or around security operations - leading a SOC, managing incident response, or overseeing detection - you've probably noticed it. The shift happened fast. Cloud adoption scaled, teams moved quicker, and security had to adapt in real time.

Your tools are in place, logs are flowing, alerts keep coming. But something still isn't connecting.

- **You Have Visibility - But Not Clarity.**
- **You See Events - But Not Impact.**
- **You Collect Data - But Don't Get The Full Story.**

This report will take a closer look at why that happens. Not in terms of criticizing what's in place, but with a focus on understanding where the current operational model falls short in a cloud-native world, and what it takes to close that gap.

Whether you're solving challenges on the ground or shaping how your organization responds to them, the ideas shared here are meant to inform, challenge, and equip you to think differently about what cloud security operations should (and could) look like.

Executive Summary

A Strategic TL;DR

Cloud operations have evolved quickly. Security operations have not.

As developers embraced cloud native architectures with dynamic infrastructure and distributed systems, security teams attempted to keep pace using legacy tools built for static environments. The result is a visibility gap. SOCs receive endless alerts but lack the context to understand what matters in the cloud, leading to missed threats and excessive escalations.

Why Traditional Tools Are Falling Behind

Tools that were built for on-prem. environments, such as SIEMs and EDRs, were extended into the cloud, but without the ability to follow real time changes or attacker behavior. Alerts are generated without insight into what changed, what was enabled, or who triggered the alert. Cloud conscious attackers intentionally exploit identity gaps and configuration drift in ways traditional tooling cannot detect.

CNAPPs Help, but Not for SecOps

Cloud Native Application Protection Platforms (CNAPPs) identify misconfigurations and surface exposure risks. But, because CNAPPs operate on static snapshots, they cannot observe real time events. Without behavior correlation or progression tracking, they miss threats that move silently through identity, network, and API layers.

The Role of Cloud Detection and Response

CDR fills the operational gap with:

- Real time, event-driven visibility across cloud layers
- Risk-based detections built on identity, network, config changes, and behavior
- Impact analysis to understand what changed in the cloud and why it matters

---

Rather than replacing existing tools, CDR enhances them with live context that turns isolated alerts into complete stories.

From Reactive to Informed Cloud Response

With CDR, SecOps teams gain the confidence to respond in the cloud with precision. For the first time, security teams will have the ability to understand cloud threat impact, mitigate and resolve cloud incidents independently, and triage alerts faster than ever before. CDR empowers security teams to defend dynamic environments with precision, not guesswork.

Stream Security was built to meet the needs of modern cloud operations.
With CDR, make security real time, actionable, and aligned with how the cloud works.

Chapter 1

How Developers Went Cloud-Native

To understand how to protect the cloud and the new, complex attack surface it brings, we must study how it was adopted. And that starts with developers and DevOps, not security. These teams didn't wait for a strategy or roadmap. They moved fast, broke things, and rebuilt them in ways that redefined how infrastructure is designed, deployed, and scaled.

Phase 1

Lift-and-Shift for Dev

In the early days, developers took what they had and moved it to the cloud. No refactoring or redesign. Just rehosting legacy apps on EC2 or other virtual machines.

That process had benefits such as:

- Faster provisioning
- Minimal infrastructure management
- No major code changes

But, it brought old assumptions into a new environment that didn't work the same way. The move to cloud was tactical, and not built for long-term strategy. But it was the first step in a much bigger transformation.

Phase 2

Replatforming and Managed Services

Once developers got comfortable with the cloud, they started to optimize. They replaced self-managed components with managed services to reduce overhead and speed up delivery.

- Databases moved to RDS, DynamoDB, or Big Query
- Object storage shifted to S3, GCS, or Blob Storage
- Authentication and permissions were offloaded to IAM systems

Outcomes

Less time maintaining infrastructure

Simpler pipelines

Faster iteration cycles

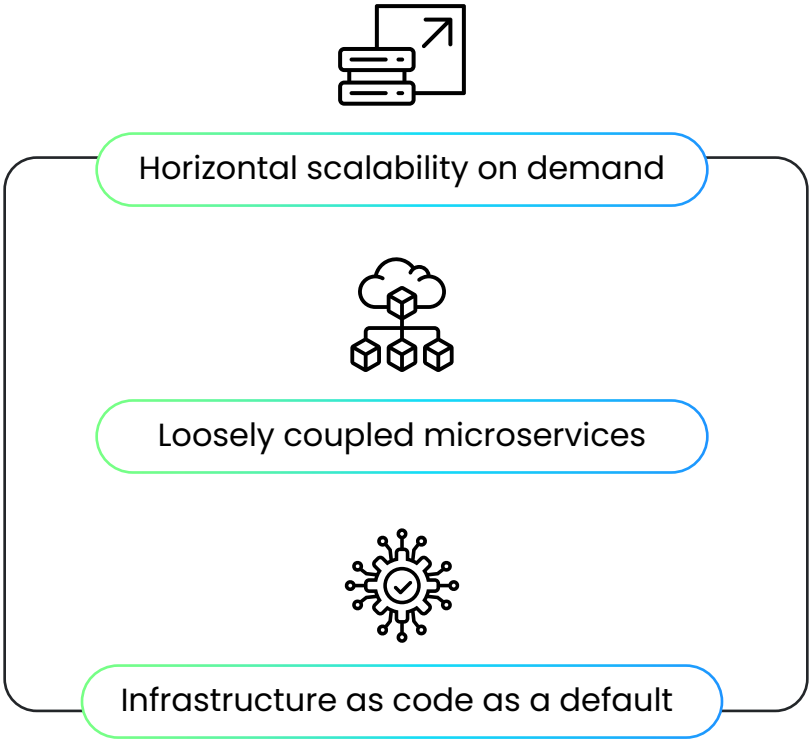
Cloud-native capabilities like auto-scaling, availability zones, and API-based provisioning became part of everyday development. Teams wrote less YAML and more business logic. But every managed service also introduced more abstraction and a growing identity surface to manage.

Phase 3

Cloud-Native Engineering

The real turning point came when teams started building for the cloud, not just in it. This marked the shift from traditional infrastructure to containerized workloads, Kubernetes clusters, serverless functions, and fully automated CI/CD pipelines.

This too brought with it several benefits in the cloud:



Engineering velocity increased significantly, deployment cycles shortened from weeks to minutes, and environments became disposable. This made cloud engineering much more scalable as infrastructure became repeatable, and teams were no longer limited by hardware or provisioning queues. At the same time, this speed brought new risks, as changes could introduce risk in seconds and disappear just as fast.

Phase 4

Multi-Cloud and Distributed Teams

Today, most engineering teams are fully cloud-native. They operate across multiple cloud providers, use best of breed tools, and build systems that are distributed, API-driven, and globally scalable by design.

Modern cloud engineering now looks more like this:

- No central perimeter
- No static infrastructure
- High dependency on managed services and third-party APIs

Cloud teams today operate in a world where infrastructure changes constantly and everything is dynamic. Teams expect full observability, real-time metrics, and zero operational drag. Unlike in its early years, cloud teams are now expected to respond instantly to business needs.

And in that world, boundaries are blurred, ownership is shared, and risk moves across systems in ways that are hard to track.

Chapter 2

The SOC Migration - Lifted Tools Left the Context Behind

When companies moved to the cloud, SecOps teams followed. But they didn't have the infrastructure to fully evolve. They lifted what they had, bringing SIEMs, EDRs, network monitoring, and vulnerability scanners to a new environment.

It made sense on paper: VMs in the cloud looked like VMs on-prem. Logs still existed. Alerts still fired. SOCs kept working the way they always had.

That illusion didn't last. With time, the cloud moved faster. It got more complex. More dynamic, more abstract. What used to be infrastructure turned into services, identities, and APIs that standard security tools couldn't see. But SOC teams kept using them anyway.

Lifting and Shifting the Toolset

Security teams extended their stack into the cloud.

- Deployed EDRs on EC2 and other virtual machines
- Forwarded CloudTrail and audit logs into the SIEM (and paid a lot for it)
- Tuned network monitoring to parse VPC flow logs
- Adapted SOAR playbooks to call cloud APIs

The shift gave SecOps teams coverage where they were comfortable: hosts, endpoints, logs. It felt like progress, but the extensions weren't built for the cloud. They were built for control. And the cloud doesn't care about control. It's built for change.

The Rise of Cloud-Conscious Attacks

Today's attackers don't just exploit workloads – they exploit how the cloud works.

Threat actors understand the architecture, know which services talk to one another, and study IAM policies, VPC routes, trust relationships, and misconfigured roles. Attackers aren't looking for a standalone app that is vulnerable. They're looking for a misstep in how everything is connected.

This is the cloud-conscious attacker.

The cloud-conscious attacker manipulates identity and network configurations to progress their attack without ever dropping malware or touching a workload.

These are some of the actions that a cloud-conscious actor might take during a breach:

- Assume a role that wasn't locked down
- Modify a security group to expose a service
- Pivot across accounts using trust policies
- Escalate privileges through chained role assumptions
- Access sensitive resources by altering VPC reachability

The breach isn't loud and won't trigger an EDR alert. But it can change an environment in significant ways. It rewires what can talk to what and who can access what. Cloud-conscious attackers move silently through environments and carry out their breaches, all while flying under the radar of traditional security tools.

This isn't theoretical. This is happening often across organizations. And if you're only watching logs or endpoints, you'll miss it.

Cloud-conscious attacks don't look like traditional threats. They look like normal activity, until it's too late.

The Risk Is in the Context - Not the Alert

SOCs don't struggle because they lack alerts. They struggle because the alerts don't tell them what matters in the cloud.

An alert says something happened but doesn't show what it means. Without details on exploitability and downstream impact, teams can't see if an alert signals the start of an attack or just noise. That's the problem.

To measure risk, the SOC needs to understand three things:

Exploitability - Is this vulnerability exposed to the Internet? Is it protected by a security control? Is it running on a critical asset or in an isolated dev environment?

Blast Radius - If this alert is real, what can the attacker reach from here? What roles can they assume? What services can they pivot to? What data is now exposed?

Activity Context - Is this event part of a bigger pattern? Is this configuration change linked to a suspicious role assumption? Is this identity showing signs of lateral movement?

Right now, most SOCs don't have access to any of that information. Their dashboard shows isolated logs and siloed alerts, showing them what happened, but not what changed. Not what a potential breach enabled.

Configuration changes often get ignored because of this exact reason – out of context, such changes can look completely harmless. Changes to security groups, role updates, or policy alterations don't trigger alerts or incidents on their own. But in the context of cloud-native attacks carried out by cloud-conscious threat actors, they can be the pivot point of a breach.

Unless you're tracking how an environment evolves (who made the change, what it affects, and what came after), these changes will never appear as threats. Without that visibility, SOCs are left guessing which alerts might matter. Oftentimes, that means they are chasing alerts that don't and overlooking the ones that do. And in the cloud, that's the fastest path to compromise.

Chapter 3

Why Security Operations Can't Just Use a CNAPP

Built for Exposure, Not Threats

CNAPPs are designed to reduce exposure. That's their job. They surface misconfigurations, flag over-permissioned roles, and highlight risky resources. They give you posture data and attack paths - but all of it is built on snapshots.

This is the core problem: snapshot-based visibility is static by design. CNAPPs sample the environment periodically, not continuously. They don't monitor what's happening. They check what things looked like. That difference matters most when attackers are already inside. Cloud-conscious attackers focused on exploiting the environment itself don't always need malware. By manipulating IAM roles, tweaking security groups, escalating privileges, and changing reachability, such threat actors can move laterally in the cloud. These are small, intentional configuration changes that fly under the radar, unless you're watching the cloud unfold in real time.

CNAPPs miss this in two key ways:

One Snapshot, Two Blind Spots

Snapshot Visibility

CNAPPs scan, analyze, and wait. What they catch is always delayed. By the time the platform identifies a risky change, the attacker has already acted - assumed roles, modified permissions, exposed services, and reached critical data. When that configuration change finally shows up, it looks benign, as the CNAPP doesn't know what came before or after.

The capture is a static artifact, disconnected from the sequence that made it dangerous.

No Event-Driven Insight

Configuration changes don't always scream compromise. Malicious intent is only revealed when connected to behavior - when you see that the public S3 bucket came after a suspicious role assumption, or that a security group was opened moments before unusual network activity.

CNAPPs by nature can't track those events because they don't correlate changes in real time. They don't follow how identities move, how services interact, or how permissions are abused across accounts.

So, while CNAPPs might tell you:

- A bucket is public
- A role is overly permissive
- A service has access to a sensitive subnet

They won't tell you:

- Who made that bucket public, and why
- Whether that role was abused minutes earlier
- If that change expanded the blast radius during an ongoing attack

Security teams mitigating breaches can't rely only on posture – they need to track progression to see the impact of actions in the cloud.

Context Without Time Is Just a Map

CNAPPs give you a map of where risks are. But they can't tell you which path the attacker took, as they lack behavior and escalation data that connects the dots. In the cloud, where infrastructure is dynamic and threats move fast, a static map isn't enough.

SecOps teams need GPS.

When the SOC investigates an alert, they need to know what changed and why it matters. But with CNAPP-based context, here's what they get:

- "This role has too many permissions."
- "This security group is open."
- "This service has access to that subnet."

What they don't get is:

- "This specific activity leveraged those permissions to escalate access."
- "This change expanded the attack blast radius five minutes ago".
- "This misconfiguration is now actively being exploited."

That difference is everything.

Chapter 4

The Cloud SOC Paradox

First to Respond. Last to Know.

In any security organization, the SOC is the first to respond. They're the ones triaging alerts, chasing down anomalies, and making the call on whether something's noise or a real threat.

But in the cloud, they're doing it without the context they need.

Despite being on the front lines, SOC analysts are often the least equipped to understand what an alert means. They see the symptoms, not the storyline. And when the environment is cloud-native, that gap becomes a bottleneck.

This is the SOC paradox.

The Frontline Blind Spot

Everything about the cloud moves fast - infrastructure spins up and down, roles change, permissions shift, and services communicate through APIs. The attacker doesn't leave malware and rewrites the rules mid-game.

SOC teams are handed alerts, but those alerts don't come with answers. They come with missing context that leaves analysts asking:

Is this role assumption a legitimate action or a privilege escalation?

Did this network change expose a workload or tighten controls?

Is this identity acting alone or as part of a broader pattern?

Without that context, two things can happen:

- SOC teams escalate too much
- They investigate too little

Analysts send alerts to cloud, IR, or DevOps teams not because the threat is clear, but because the data is incomplete. They burn time chasing noise or miss what matters altogether.

Context at the Detection Phase - Not After

By the time an alert hits the SOC queue, it's already too late to go piecing together logs, IAM policies, and VPC flow data. The context must be give up front, upon detection.

What does that look like?

- A full view of the attack storyline, not a single event
- Immediate insight into what changed, who did it, and what it exposed
- Clarity on blast radius, exploitability, and service ownership
- Confidence in what to escalate, what to contain, and what to ignore

Flipping the SOC paradox means moving security analysts from reactive triage to informed response, from chasing alerts to understanding threats so that teams can be operational, not overwhelmed.

Chapter 5

Cloud Detection and Response

Real-Time Context. End-to-End Storyline. Finally.

Cloud detection and response (CDR) didn't emerge as another tool category. It emerged out of necessity. Every existing approach failed to keep up with how the cloud works.

CDR is not posture, log correlation, or a dashboard for risks that you might want to remediate next quarter.

CDR is built for the here and now - to give SOC teams real-time clarity in cloud environments they don't own, don't control, and often don't fully understand.

The Core Principle: Real-Time Context Across Every Cloud Layer

CDR is grounded in one idea: you can't detect what you don't understand, and you can't respond to what you can't see in motion.

To do that, CDR captures real-time context across every layer of the cloud - not just logs, not just snapshots, but live awareness of how the environment behaves.

Network

- Understand what's allowed (VPC configuration, routes, security groups)
- See what's actually happening (east-west traffic, API calls, service-to-service communication)

Identity

- Map static permissions across all roles and accounts
- Monitor identity behavior - who is assuming what, when, and how it moves laterally

Kubernetes

- Track activity inside clusters - pod behavior, API activity, RBAC misuse
- Correlate K8s events with surrounding cloud activity

Threat Intelligence

- Enrich detections with known attacker infrastructure, IOCs, and cloud-focused TTPs
- Detect abuse that mimics real attacker patterns, not just theoretical misconfigurations

Exploitability

- Combine vulnerabilities, sensitive data, and exposed services into one clear model
- Prioritize based on what an attacker could actually reach, not what's just technically present

CDR provides a live model of your environment that is updated in real time and designed to surface not just activity, but what that activity means in the bigger picture.

Why Impact Analysis Changes Everything

CDR doesn't just say, "This role was assumed."
It answers:

- What could that role access?
- Was that access used?
- Did it lead to a configuration change or data exposure?
- What was the downstream effect?
- Who owns the affected resource?

Every action, every change, every movement is assessed for impact.

Because that's what makes an alert meaningful - not that it happened, but that it changed something, whether that's advancing an attack or shifting risk profile in real time. CDR surfaces real threats while ignoring the noise.

The Storyline the SOC Never Had

Traditional tools give alerts, while CDR gives storylines.

Instead of a dozen disconnected events, CDR connects the dots:

- A role was assumed
- ↓
- A security group was modified
- ↓
- A workload was accessed
- ↓
- A lateral move occurred
- ↓
- Sensitive data was exfiltrated

CDR ties this data together in real time, with full context – even for analysts who don't speak fluent cloud. CDR doesn't expect SOC teams to understand cloud infrastructure, translating cloud-native behavior into investigative clarity. It gives them the "what," the "how," and the "why" - instantly.

Chapter 6

CDR Isn’t a Replacement

It’s the Missing Layer That Makes Everything Else Work

There’s a misconception that bringing in a CDR platform means overhauling your stack, ripping out tools, and replacing what you already rely on.

That’s not the point of CDR.

CDR doesn’t replace your EDR, SIEM, or vulnerability scanner. It integrates with them, giving them the missing context that makes them useful in the cloud.

CDR fills the real-time visibility gap that every other tool was never designed to cover. It connects the dots between what your stack sees and what’s happening across your cloud.

CDR Is the Context Engine for Your Stack

Think of CDR as the real-time brain that feeds your existing systems with cloud-native understanding. Here's how it fits in:

EDR: From Local Events to Cloud-Driven Insight

EDR is still critical for workload-level protection. But EDR on its own has no idea what's happening in the cloud around the workload.

CDR integrates by enriching EDR alerts with cloud context:

- Was the machine compromised after a role assumption?
- Did a misconfigured bucket or exposed secret lead to this incident?
- What's the blast radius based on surrounding cloud services?

> With CDR, endpoint detection and response becomes a signal in a broader cloud attack storyline rather than an isolated endpoint alert.

SIEM: From Raw Logs to Actionable Impact

SIEMs are still useful for collecting and storing logs across the enterprise. But in the cloud, logs without context are noise.

CDR doesn't replace your SIEM - it gives it meaning to its data, allowing the SIEM surface alerts with full cloud-aware context: Not just "what happened," but "what was the impact".

SIEM alerts enriched with cloud context show:

- Not just "who logged in," but "what they accessed and changed"
- Not just an event stream, but a cohesive attack storyline

You still ingest the logs - but now they tell you something that matters.

Vulnerability Management: Contextual Exploitability

You already know where the vulnerabilities are. But which ones matter?

CDR integrates with your vulnerability scanner to understand exploitability in real time:

- Is this CVE reachable through a misconfigured role or open port?
- Is this vulnerable asset exposed to the internet or just internal?
- Is there lateral movement that makes this now-internal risk urgent?

This shifts vulnerability management from static risk scores to operational urgency.

DSPM: Sensitivity Meets Threat Activity

CDR ties into your Data Security Posture Management (DSPM) platform to layer data sensitivity on top of cloud activity:

- Is this role touching sensitive data for the first time?
- Was data accessed following a privilege escalation or lateral move?
- Is a workload interacting with PII or regulated assets in an unusual way?

Now alerts aren't about behavior. They're about what's at stake.

eBPF: Runtime Visibility in Kubernetes

Kubernetes is a blind spot for most SOCs. CDR integrates with eBPF agents to track runtime activity inside clusters, including:

- Container-level behavior
- API calls between pods
- Misused RBAC or unusual service communications

This runtime telemetry is correlated with cloud-level identity and network context, giving the full picture from workload to environment.

Threat Intelligence: Detecting IOCs in Motion

CDR consumes threat intel feeds not as static indicators, but as dynamic inputs to detect known malicious IPs interacting with your services, suspicious DNS requests tied to active role usage, and cloud behavior matching known attacker TTPs. This allows teams to move past matching logs and towards understanding threat behavior in context, across the entire cloud stack.

The Bottom Line

CDR doesn't replace what you've built. It completes it.

It connects your EDR to identity abuse. Your SIEM to actual impact. Your vulnerability data to real exposure. Your DSPM to threat progression. Your eBPF runtime to cloud-layer behavior. Your intel feeds real detections.

CDR is the layer that finally ties it all together - so you're not staring at disconnected alerts but seeing the full story.

In the next chapter, we'll show what that looks like in action and how CDR reduces time-to-detect, time-to-triage, and time-to-understand in ways traditional tools can't.

Chapter 7

This Is What Cloud-Native Security Looks Like

Cloud security is entering a new phase, where the focus is shifting from chasing alerts to influencing how the cloud operates.

Cloud Detection and Response supports this evolution by giving teams real-time control, clarity, and confidence.

The role of cloud-native security extends beyond protecting infrastructure. It helps the business move faster with fewer obstacles. CDR gives the SOC the ability to act with precision rather than guesswork.

Where CDR is Going Next

1 Real-Time Guardrails That Guide and Protect

CDR helps teams build safely with real-time enforcement: flagging and correcting violations the moment they happen.

Whether it's a misconfigured IAM role, a risky open port, or a policy drift that exposes a critical resource, the CDR tool responds instantly, before attackers ever have a chance to exploit it.

2 Response That Spans Every Layer

With CDR, response becomes more than reactive containment - it becomes orchestration. Security teams can:

- Roll back a risky configuration
- Kill a live session that's stepping out of bounds
- Apply guardrails that adapt as the environment evolves
- Quarantine only what's needed, without disrupting the rest

The value here isn't just automation. It's precision. It's knowing what to respond to - and how - with full awareness of the cloud's state and context.

3 Understanding the Business Impact Before You Act

Every action has consequences. And security can no longer afford to operate in isolation.

CDR helps you predict the ripple effect of every response:

- Which services will be affected?
- Who owns this resource?
- Will this action break a production workflow or block critical access?

To strike the balance between risk reduction and business continuity in cloud threat response, security teams need full, real-time context.

Empowering the SOC to Lead, Not Follow

When SOC teams are armed with the full picture, not just the symptoms, they stop being the bottleneck. They become enablers of safe innovation.

They make better decisions.

They investigate less but understand more.

They stop escalating every unknown - and start resolving what matters.

CDR gives them the ability to lead, response, and trust the environment they're defending.

This Is How You Do Security in the Cloud

Security in the cloud is not done with static tools built for another era, based on stitched together visibility and siloed alerts.

SecOps teams need a platform that models the cloud the way it actually behaves - fast, dynamic, layered, and alive.

This is what cloud-native security operations looks like.
This is how you empower teams to move with speed and security side by side.
This is how you adopt innovation without giving up control.

And the Best Part?

It's 2025. And you've just read an entire report about the future of cloud security - real-time, contextual, deeply integrated - without once using the word "AI."

Because this isn't hype. This is clarity.
The cloud is already moving at full speed.
Now security finally can too.

Stream Security: Built for This Moment

At Stream Security, we pioneered Cloud Detection and Response to give security teams exactly what the cloud demands: real-time visibility, deep context, and the ability to act with precision. We model your environment as it evolves, uncover attack storylines as they unfold, and give your SOC the clarity to respond in seconds, not hours.

Whether you're operating at global scale or navigating complex multi-cloud architectures, Stream Security helps you align cloud speed with enterprise-grade security - without compromise.

Ready to see what real-time cloud security looks like?
Learn more at stream.security.