

# The Detection Engineering Breaking Point

## How Agentic AI Changes the Equation

### Why SOC Pain Points Flow Upstream to Detection

For years, security operations leaders have been pushing a simple but powerful idea: **shift detection engineering left**. Treat detections as code, manage them through lifecycle processes, map SIEM and EDR rules to adversary behaviors, then continuously tune, validate, and refine them.

In theory, this approach transforms the SOC. Instead of reactive alert triage, organizations build a structured detection program that systematically identifies attacker behaviors and produces high-fidelity alerts. Detection engineering becomes the control plane for threat detection.

Over the past decade, the industry has made real progress toward that vision. Detection-as-Code, ATT&CK frameworks, modern SIEM platforms, and modular security data architectures have all helped push the SOC in that direction.

This philosophy is spot on, but the **operational reality required to execute it at scale deteriorates faster than teams can keep up**. The pains the SOC experiences—alert fatigue, missed threats, inconsistent coverage—can be traced back upstream to one place: The detection layer.

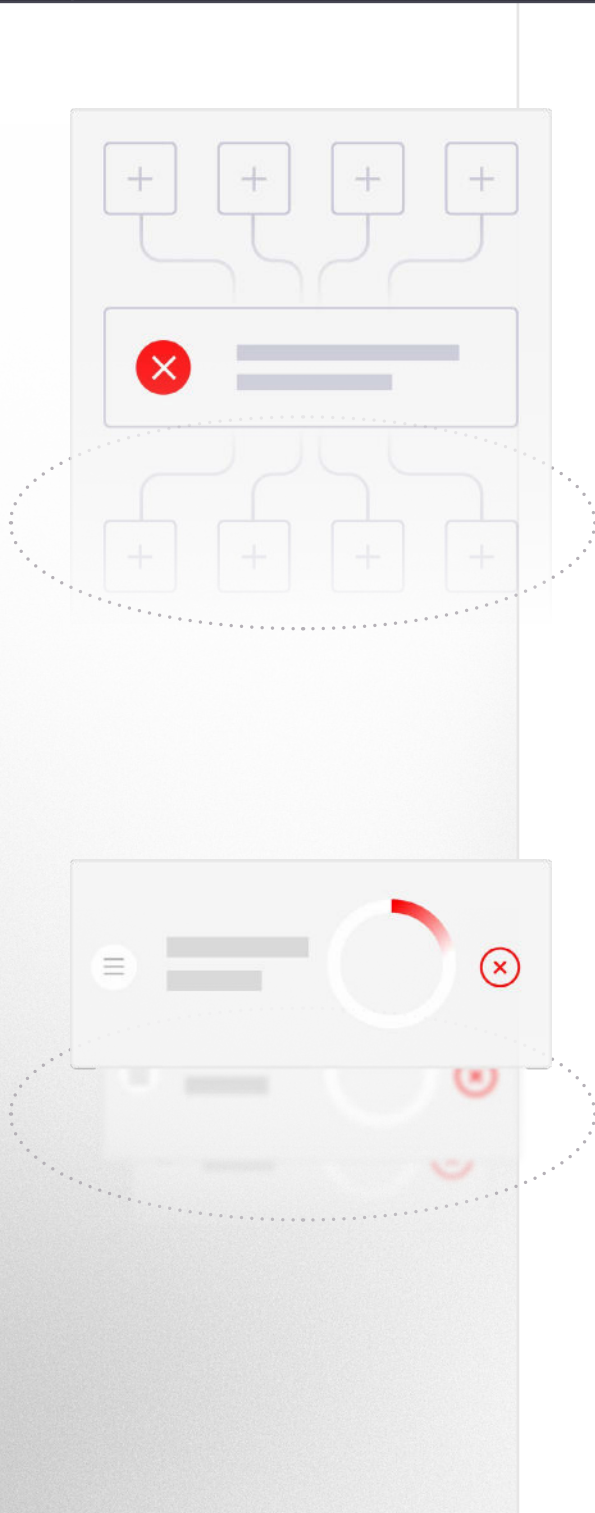


# The Growing Pressure on Detection Engineering

Modern SOC problems rarely begin with analysts drowning in floods of alerts.

They begin in the detection programs that eventually create alerts.

Detection engineering sits at the center, absorbing pressure from several converging forces.



## Exploding Telemetry

Enterprise environments now generate security telemetry at a scale that early SIEM architectures never anticipated.

Cloud workloads, container orchestration platforms, SaaS ecosystems, identity providers, endpoints, APIs, and custom applications all generate logs and events that contain valuable detection signals.

Every telemetry source presents opportunity AND responsibility. Each one typically requires parsing and normalization, data quality validation, detection logic development, coverage mapping, tuning and lifecycle maintenance.

The number of signals for detection engineers to consider is growing faster than the teams responsible for managing them.

## Faster, More Adaptive Adversaries

At the same time telemetry is expanding, attackers are evolving faster than ever.

Adversaries are increasingly leveraging automation and AI to accelerate their own operations, with rapid tool iteration, obfuscated command execution, identity-centric attack paths, living-off-the-land techniques, and infrastructure churn for defense evasion.

These tactics produce subtle behavioral signals, not obvious indicators. That forces detection programs to continuously evolve their coverage of attacker techniques with dynamic approaches.

## Detection Stack Complexity

The detection stack itself has also grown significantly more complex. Modern environments typically include:



Multiple SIEM  
or analytics platforms



Endpoint  
detection tools



Security  
data pipelines



Security analytics  
processing systems

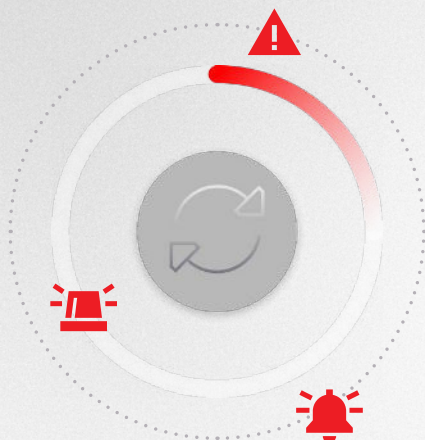


Data lakes  
and cold storage



Threat intelligence  
platforms and feeds

Detection logic often spans all of these systems. A single rule may depend on upstream data pipelines, normalized schemas, enrichment processes, and scheduled queries across multiple platforms. As environments grow more distributed, detection logic becomes more complex, harder to validate, and easier to break.



## Noisy Alerts and Broken Feedback Loops

The consequences of this complexity are felt most acutely by SOC analysts. Poorly tuned detections generate false positives that overwhelm analysts, who then struggle to provide meaningful feedback.

Analysts are closest to the operational reality of detections, but their insights rarely flow back effectively into the detection engineering process. Meanwhile, detection engineers are buried in rule maintenance and new detection development. The SOC experiences the symptoms. Detection engineering carries the root cause.

# The Manual Workload Behind Detection Programs

The promise of detection engineering is powerful, but the actual day-to-day work required to sustain it is enormous. Mature programs eventually encounter some form of the following operational burdens.

## Detection Rule Maintenance and Technical Debt

Over time, detection rule sets count into the hundreds, or even thousands. Each rule carries hidden dependencies: specific event fields, parser logic, data pipeline transformations, scheduled execution parameters, and platform-specific query syntax.

As environments evolve, these dependencies break. Platforms update. Schemas change. Fields disappear. Pipelines drift.

Without continuous validation, detections silently fail or begin generating unreliable alerts. Rule sets slowly accumulate **technical debt**, and engineers spend increasing amounts of time maintaining old detections rather than building new ones.

## Adversary Mapping and Coverage Management

Many organizations map detections to frameworks like **MITRE ATT&CK** to understand coverage against attacker behaviors.

This sounds straightforward in theory. In practice it requires maintaining a continuously updated inventory of which rules exist, which adversary techniques they cover, which telemetry sources they rely on, and which techniques remain uncovered.

As rule sets grow and environments change, maintaining this mapping becomes a major operational task. Without it, coverage programs quickly degrade into vanity dashboards that imply coverage but lack operational rigor.

## Investigations of Noisy Detections

Alert fatigue rarely appears overnight. It creeps in gradually as detections drift away from the environment they were designed for.

**Detection engineers frequently find themselves investigating questions:**

- ④ Why is this rule firing hundreds of times per day?
- ④ Which field values are driving these false positives?
- ④ Did a telemetry source change?
- ④ Is the parser broken?
- ④ Is the logic flawed?

Answering those questions requires combing through alerts, raw logs, and rule logic across multiple systems. The tuning process can take hours or days for a single detection.

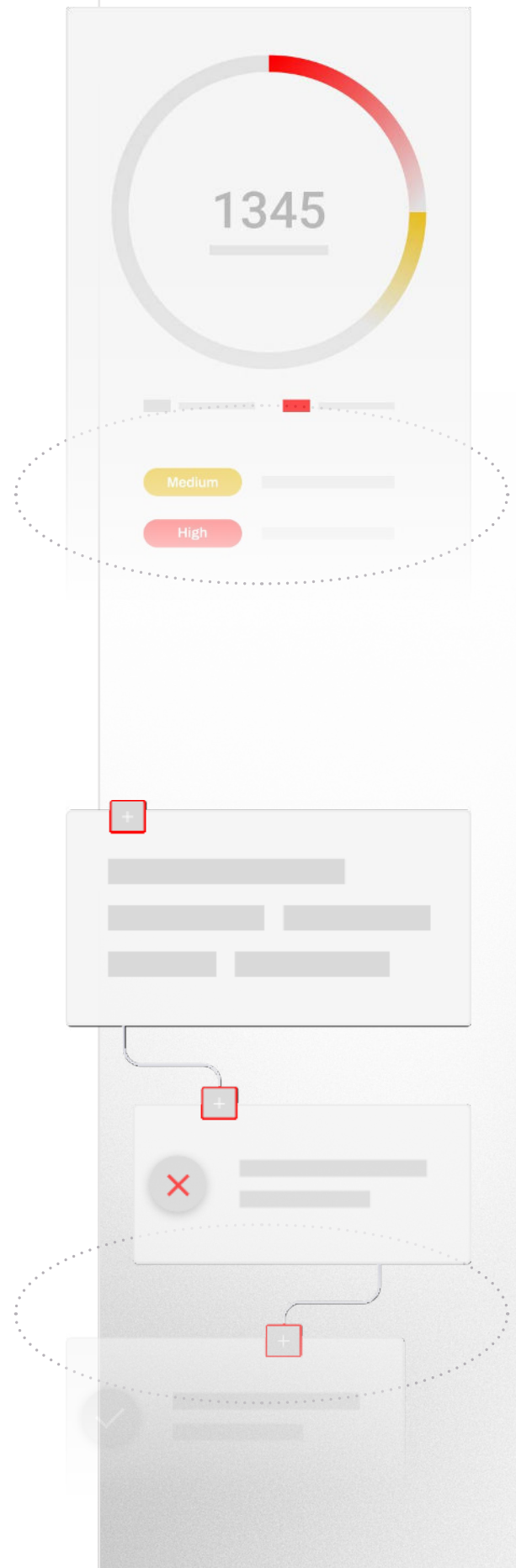
Multiply that across hundreds of rules, and the workload becomes enormous.

## Threat Hunting and Detection Calibration

Detection engineers are rarely responsible only for maintaining rules, also participating in threat hunting. That work includes generating hypotheses about adversary behaviors, exploring telemetry to identify potential signals, and building new detections based on discovered patterns.

It is some of the most intellectually valuable work in the SOC, but it competes directly with the operational burden of maintaining existing detections.

The result is a familiar problem: Detection engineers get spread too thin across too many responsibilities, constantly context-switching between innovation and maintenance.



# Why the Current Model Struggles to Scale

Put all the above pressures together, and a pattern begins to emerge.

Detection engineers are asked to do incredibly heavy lifting with limited headcount and budget constraints. Experienced practitioners are in short supply, so many organizations find themselves caught between two uncomfortable realities: they have more telemetry than ever but cannot reliably convert it into high-fidelity detections.

The shift-left vision is correct, but the prevailing operational model has not kept pace with the scale of modern environments.

## The Emergence of Agentic Detection Engineering

The breadth and depth of detection engineering skills span interpreting adversary behaviors, understanding complex enterprise environments, and making strategic detection decisions that require advanced contextualized reasoning. Doing it at a high level will continue to fall on experienced practitioners.

But the operational burden surrounding detection programs—the analysis, maintenance, and pattern discovery across large rule sets—is precisely the type of problem agentic AI systems are well suited to address.

In this emerging **Agentic Detection Engineering** model, detection engineers remain the strategic architects of detection programs.

AI agents operate as **continuous collaborators**, helping teams:

- ✓ Analyze and continuously expand detection coverage
- ✓ Eliminate alert noise and identify rule failures
- ✓ Surface new detection opportunities
- ✓ Connect telemetry with adversary behaviors

**The goal is never to replace detection engineers but to enable them to be more effective and strategic.** Instead of drowning in maintenance tasks, engineers can focus on the work that actually improves security outcomes: understanding adversaries and designing meaningful detections.

# Introducing the CardinalOps Agentic Fleet



**False Positive Terminator** (Score: 96)

- Core Skills:**
  - 96 - Contest Intelligence
  - 95 - Alert Efficiency
  - 90 - SOC Sanity Preserved
- Special Ability:**
  - TOTAL PURGE:** Kills noisy false positives upstream, injects context from asset to alert.

**Threat Hunter** (Score: 97)

- Core Skills:**
  - 98 - Adversary Induction
  - 98 - Threat Intel
  - 98 - Hypothesis Generation
  - 98 - Tolerant Infiltration
  - 97 - Threat Intelligence
  - 95 - SOC Sanity
- Weapons:**
  - Attack Path Hypothesis Engine
  - Threat Intelligence Correlation
  - Tolerant Gap Discovery
  - Detection Opportunity Generation
- Special Ability:**
  - GHOST TRAC:** Exposes attack movement that previously left footprints.

**Detection Synthesizer** (Score: 93)

- Core Skills:**
  - 93 - Signal Reduction
  - 93 - Alert Reduction
  - 93 - Efficiency
  - 93 - Tolerant Infiltration
  - 93 - SOC Sanity
- Weapons:**
  - SIEM Rule Overlay Analyzer
  - Redundancy Correlation Engine
  - Detection Lifecycle Manager
  - Auto-Optimization Pipeline
- Special Ability:**
  - CLEAN SWEEP:** Weeps, wails, and rebuffs your detection stack from the inside out. No rule left unscathed.

**Coverage Surveyor** (Score: 94)

- Core Skills:**
  - 96 - ATTCK Mapping
  - 95 - Coverage Gap Radar
  - 94 - Detection Posture Scoring
  - 93 - Adversary Alignment
- Weapons:**
  - ATTCK Mapping Engine
  - Coverage Gap Radar
  - Detection Gap Radar
  - Detection Posture Scoring
- Special Ability:**
  - ATTCK ATLAS:** Maps every detection rule to real adversary behavior and exposes the gaps attackers exploit.

Advances in AI and agentic capabilities present a real inflection point. Not simply through generative assistants that help write queries, but through **agentic workflows that operate continuously across detection programs.**

That's why we're introducing the CardinalOps Agentic Fleet. It's less about chatbots and more about AI teammates embedded inside the detection lifecycle. These agents can operate across detection programs at scale and speed, addressing these pain points in unique ways.

Instead of relying solely on manual effort, it introduces a coordinated system of specialized AI agents that optimize the entire detection lifecycle and fly alongside human detection engineers.

Each agent focuses on a critical domain—signal quality, coverage, efficiency, and threat discovery—operating continuously across telemetry, detection rules, and alert outcomes. Together, they transform detection engineering from a reactive, resource-constrained function into a scalable, adaptive, high-performance system.

Detection teams can scale without adding headcount. Alert fatigue gives way to signal clarity. Detection lifecycle management becomes streamlined instead of sprawling. And perhaps most importantly, feedback from the SOC finally closes the loop—feeding directly back into better detections over time.

# Meet Your Fleet Operators

Each agent in the fleet is purpose-built, with a clear objective and a distinct role in strengthening detection programs. Individually, they solve persistent challenges. Together, they operate as a coordinated system.

## False Positive Terminator



Noisy alerts are more than an annoyance—they're a tax on the entire SOC. This agent continuously analyzes alert patterns and historical triage outcomes to identify which detections are generating excessive noise. It goes beyond just flagging the problem. It pinpoints root causes, recommends precise tuning strategies, and provides clear reasoning.

With human review and approval, it implements targeted improvements that reduce false positives while preserving detection integrity. The result is a measurable drop in alert fatigue, faster response and investigation cycles, and a detection program that learns from real-world outcomes instead of drifting away from them.



## Threat Hunter

While some agents focus on what's readily apparent, the Threat Hunter dives into the unknown. It continuously ingests threat intelligence and indicators of compromise, searching across your environment to surface suspicious patterns and theorize on potential attacker activity.

It generates hypotheses about how adversaries might operate in your environment and identifies signals worth turning into new detections. This transforms threat hunting from a time-intensive, manual effort into a continuous, scalable capability that helps teams reduce dwell time and stay ahead of emerging tactics.



# Meet Your Fleet Operators

## Detection Synthesizer



Detection environments inevitably accumulate redundant rules overlapping logic, and outdated detections. This sprawl adds unnecessary complexity and cost to the detection stack. The Detection Synthesizer cuts through the noise by analyzing rule sets for duplication and inefficiency.

It identifies opportunities to merge, optimize, or retire detections, then recommends lifecycle improvements that simplify the entire detection stack. With approved changes implemented automatically, teams benefit from reduced operational overhead, lower compute costs, and a cleaner, more maintainable detection program.



## Coverage Surveyor

Knowing your current detection coverage is part of the battle, but understanding your most critical blind spots creates the most powerful risk mitigation opportunities. The Coverage Surveyor continuously maps detections to adversary behaviors, identifying gaps across tactics and techniques while surfacing missing telemetry sources to close those gaps.

It evaluates available telemetry, recommends where new detections can be built, and prioritizes gaps based on risk. This turns coverage management from a static reporting exercise into a living, evolving strategy—ensuring detection programs stay aligned with both the threat landscape and the environment itself.



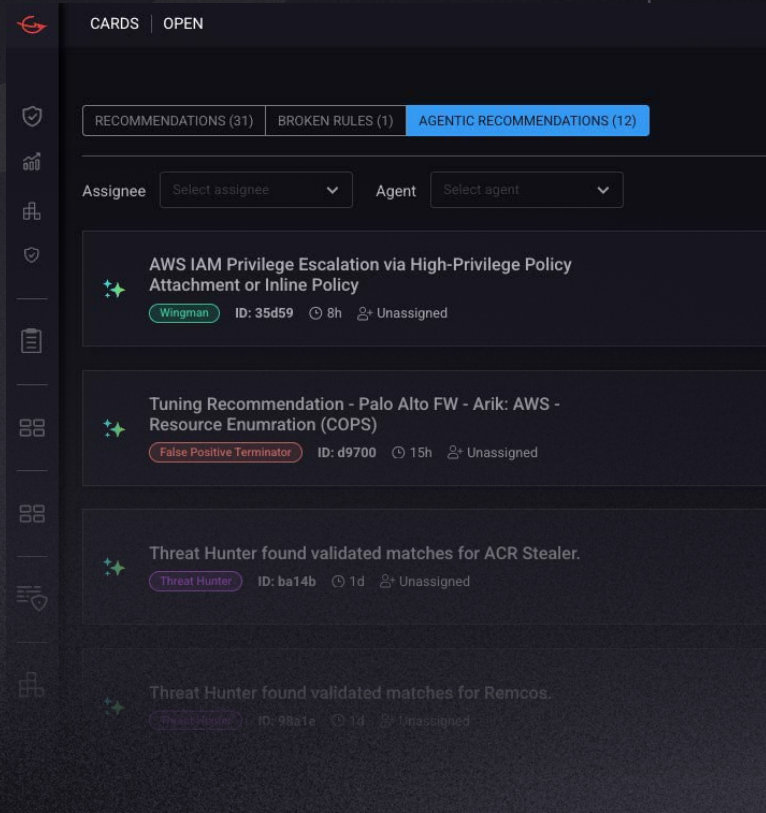
# A New Chapter for Detection Programs

Detection engineering transforms telemetry into meaningful security signals. But as environments have grown more complex, endless manual work has made the discipline increasingly difficult.

The next evolution of the SOC depends on augmenting detection engineers with intelligent systems that operate continuously across detection programs.

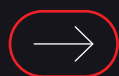
The core CardinalOps platform enables that evolution, unlocking the full potential of detection engineering with AI-powered workflows that continuously optimize coverage. Our new Agentic Fleet takes that one step further, helping engineers reason over telemetry, rule logic, adversary behaviors, and alert outcomes at a much broader scale.

Organizations that embrace this new model will be prepared to continuously adapt to both evolving environments and evolving adversaries. And that may ultimately be the only way modern SOCs keep up.

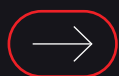


## Change the Detection Engineering Equation

Detection engineering shouldn't feel like an uphill battle. If your team is buried in rule maintenance, chasing false positives, and struggling to keep coverage aligned with reality, let's talk. We'd love to review your detection program and discuss AI-powered workflows that continuously improve signal quality and strengthen coverage, so you never miss another threat.



**Contact Us**



**cardinalops.com**



**CARDINALOPS**