expel

Which signals matter?

A security leader's guide to high-fidelity detections

CO-AUTHORED BY:

Jeff Archer, Senior Detection Response Engineer

Brady Stouffer, Manager of Detection and Response Engineering

Sarah Crone, Senior Product Marketing Manager

The alert queue is broken (and it's not your fault)

Another day, another thousand alerts. Your SIEM is firing off alerts, your EDR is having a meltdown over PowerShell executions, and your cloud security tools are convinced every API call is suspicious. Meanwhile, management keeps asking why you need more headcount when you already have "all these advanced AI-powered detection tools."

Sound familiar?

This guide is for security leaders and managers who are looking to uplevel their detection strategy. It was written in partnership with real Expel practitioners and detection engineers who understand what it's like to work a SOC at 2 a.m. This isn't another vendor pitch disguised as thought leadership. This is a practical guide written for the people actually doing the work—SOC analysts, managers, and directors who need to separate signal from noise without tripling their budget or their team size.

Let's get into it.

Why boasting "more detections" is marketing BS

You may have chosen your security vendors thinking they would help you write detections. But almost every security vendor loves to brag about their detection and integration count like it's a high score in a video game. "We have 10,000 detection rules!" Great. How many of those actually matter? How many are duplicates? How many fire on every Windows update?

Think about it:

- **Vendor A** has 1,000 rules and generates 500 alerts a day, with a 1% true positive rate. That's 5 real incidents buried in 495 false alarms.
- **Vendor B** has 100 rules but a 10% true positive rate at the same volume. That's 50 real incidents with 450 false alarms.

Which vendor would you rather work with? While 450 false alarms still seems like a significant number, it's a huge improvement compared with vendor A.

You need actionable detections that are built for your specific environment. For example, a tool claiming to protect against a variety of PowerShell attack types is great only if you have a Windows environment. What good is that high number of detections if none of them are relevant to your context? The key isn't a perfect solution, but one that reduces the workload on your team. More detections can give you a false sense of security while a real threat slips through the cracks.

Alert fatigue is a system problem, not a people problem

It's a predictable human response to become overwhelmed by low-quality information. When you're processing 200+ alerts per shift, your brain starts taking shortcuts. You develop heuristics: "This alert type is usually nothing," or "This system always cries wolf."

These shortcuts aren't laziness—they're survival mechanisms. Attackers know this. They count on their malicious activity getting lost in the noise of your poorly tuned security stack. This problem only compounds when senior analysts train new team members. When they're teaching new analysts, they tend to pass along their biases, and it creates a dangerous cycle where institutional knowledge becomes about what to ignore, rather than what to investigate.

The solution isn't more people or more caffeine. It's better detections and partners who understand this.

Moving from alerts to actionable detections

Most security tools generate alerts, aka isolated events that require an analyst to play detective across tools and attack surfaces to discover a threat. Real detections provide actionable insight, giving you enough context to make a confident decision.

Let's get our terms straight. In a world of noise, there are four core concepts to define:

- **Events:** A raw observation from a system, like "PowerShell spawned a new process." On its own, this is just a data point—it has no inherent security disposition. We treat external data as events until we determine otherwise.
- **Signals:** The meaningful, security-relevant information you extract from those raw events. This is the crucial step of filtering the noise for something real.
- **High-fidelity detection:** The enriched, correlated insight transforming multiple signals into probable threat identification. This sits between raw signals and final alerts, providing the engineering work to connect the dots.
- **Alerts:** A clear, actionable disposition that something is malicious, policy-violating, or undesirable. This is what's worthy of your team's time.

So when you're thinking about detections versus alerts, consider this. A detection is something that emits an alert. And an alert should be more than a detection simply executing. It must articulate what's wrong, what needs changed, what to investigate, and where to continue monitoring. A detection is like a motion sensor, and an alert is like the camera capturing the motion from every angle, ready to play back each still frame as needed.

Most tools stop at the event or signal level and just blast you with volume. Your team then has to manually perform the "engineering" to correlate these low-context data points. A **high-fidelity detection** is the final product of that engineering work. Here's the difference in a real-world example:

A typical alert	High-fidelity detection with full story
The PowerShell process spawned with suspicious parameters on DESKTOP-ABC123.	DESKTOP-ABC123 (Finance dept, assigned to J. Smith) executed a PowerShell script launched from a malicious email attachment. The script established a command and control connection to known threat actor infrastructure, following a suspicious VPN login from Romania 10 minutes prior. J. Smith's normal location is Chicago. Confidence: High. Recommended action: Isolate the endpoint and reset the user's credentials.

Instead of receiving a hundred individual signals about a user's activity, you get one coherent, actionable alert that says, "Here's the threat and here's the full context." It's the difference between playing detective and being given the answer. The goal is to get to the second kind of detection, not by adding more tools, but by making your existing ones work smarter.

A practical framework for better detections

So far we've talked about what a good signal looks like, but let's shift to some actionable steps you can take to build out a good detection strategy. Our own detection engineering team relies on a framework we'll share with you. This is a straightforward approach based on what works in real-world SOCs, not what looks good in a demo.

1. Purposeful ingestion: Stop drinking from the fire hose

Before you ingest another log source, ask a simple question: Will this make my life easier or harder? A lot of log sources just make your life harder without adding value. They add volume without giving you additional context or clarity.

A basic security tool might produce thousands of alerts for every single true positive—a lead alert percentage of effectively zero. On the other hand, a well-tuned integration might have a lead alert percentage of 1-2%, meaning for every 100 alerts, one is a true, actionable incident. This is a massive difference in efficiency.

This means you need to get strategic and start with fundamental questions like, "What pain point or what gap are we trying to address?" Or "How is this data or new tool going to help us, and what's the minimum we can get out of this tool to help us?" Before you onboard a new data source, conduct a value analysis on your integrations. It's a simple, pragmatic exercise:

1. **Define the potential value:** What's the threat this data will help you see? Does it provide context you're currently missing? Will it help you turn a low-confidence signal into a high-fidelity detection?
2. **Assess the cost:** The cost isn't just about money, it's about time and compute resources, too. How much engineering time and effort will be required to tune and maintain detections for this source? What's the cost of storage and compute? How much analyst time will be required to triage the alerts? Will the detections it offers fill a void in your MITRE ATT&CK coverage?
3. **Calculate the ROI:** Compare the value to the cost. If a data source offers minimal insight with no correlative value (but creates a ton of low-confidence alerts) it's not worth ingesting. It'll just add to your team's burnout and dilute signals that actually matter.

Certain signals are more valuable than others. So, instead of trying to make sure all vendors have the exact same amount of detections or coverage, focus on what's *actually* going to be impactful.

2. Consistent normalization: Make your tools speak the same language

Raw alerts are useless without context. The key is normalization and enrichment—making sure every alert is standardized and enriched with the same contextual information. Think of it as a translator for your security stack. The goal here is to make every piece of data speak the same language, regardless of its source.

Imagine an alert for a suspicious file on a laptop and another for a cloud storage access event. In a fragmented environment, these are two separate, unrelated clues. But with a consistent normalization framework, those events are both enriched with vital context like:

- **Asset ownership and value:** The laptop belongs to a senior executive; the cloud storage contains sensitive financial data.
- **User roles:** The user is an administrator with privileged access.
- **MITRE ATT&CK mapping:** The activity maps to specific attack techniques, like Lateral Movement (T1021) and Exfiltration (T1041).
- **Behavioral baselines:** The access is anomalous because the executive has never accessed this storage bucket before.

This normalization process transforms events into signals and gives your team the confidence to act. When every event is consistently tagged and enriched, correlation engines can automatically connect related activities into high-fidelity detections. You're effectively building a common data schema that empowers your team to automatically connect the dots..

This isn't just theory. It's the difference between a 30-minute investigation and a 30-second decision.

3. Confident action: Stop creating tickets for everything

Not every alert needs a ticket. The action you take should match the confidence level of the detection.

- **Low confidence, isolated signal:** Log and monitor automatically through pattern analysis and threshold monitoring–there is no need for human involvement. The right action may be to simply update a dashboard and log it for automated trend analysis. Think of this as the initial "sniff test" performed by a machine.
- **Expected activity:** Use intelligent suppression. A great example is a security researcher performing a penetration test. The system recognizes the activity, tags it as benign, and suppresses the alert so it never reaches an analyst's queue.
- **High confidence detection:** Deliver a pre-built investigation package to an analyst with all the necessary context and recommended actions. The system does the heavy lifting—correlating all the disparate signals, enriching them with context, and providing the analyst with a single timeline of events. The analyst receives a complete picture, a high-confidence assessment rather than a clue, allowing them to focus on containment and remediation rather than spending hours on manual triage.

Your human talent is your most valuable resource. Use it for what actually matters.

The anatomy of a high-fidelity detection: Breadth, depth, and precision

So you want a detection program you can actually count on? Having a solid operational framework is necessary, but it's not enough on its own. The quality and scope of the detections themselves are what separate a great security operation from an average one. This requires a focus on three dimensions of coverage: breadth, depth and precision.

Breadth: Seeing the entire battlefield through a connected ecosystem

You can't detect what you can't see. True breadth is about more than just having many integrations; it's about technological freedom and comprehensive visibility. The common argument you hear is that a single-vendor ecosystem is better because everything "just works together." In reality, this often locks you into a single vendor's blind spots and forces you to rip and replace technologies you've already invested in. Or worse, the "single-vendor ecosystem" is just a patchwork of acquired technologies hastily bolted together, resulting in a frustrating user experience.

A mature detection strategy leverages the best tools for the job, the ones you've already invested in, without forcing you into a single vendor's ecosystem. This means finding a central hub (could be a SIEM, XDR, MDR platform, or even a data lake) and taking an API-first approach, allowing you to seamlessly integrate with your existing security stack. This will allow you to integrate with a vast ecosystem spanning across all of your attack surfaces.

Depth: Understanding the attacker tradecraft, not just their tools

When it comes to detections you need deep knowledge of attacker techniques for every attack surface. The out-of-the-box detections you get from your vendors are great for providing a baseline, but they are generic and well-known to adversaries.

A truly resilient program focuses on the underlying attacker technique (TTP), not just the indicator (a file hash or IP address). For example, an attacker might compromise an account and then use a known tool to perform "credential dumping." The file hash of

the tool (the indicator) will change tomorrow, but the technique, the process of dumping credentials, will not. A detection focused on the TTP is far more resilient. This requires a human-driven (with analysts developing and refining detections) and machine-accelerated (with AI/ML tools building dynamic baselines) detection strategy.

Precision: Connecting the dots across surfaces

Precision is the operational outcome of layering breadth with depth. It's how you measure your ability to transform thousands of low-confidence signals into a handful of high-confidence incidents. This is where the magic happens. By automatically correlating related events across multiple surfaces into a single, cohesive timeline, a precision-focused detection program eliminates the manual, time-consuming investigation slowing down mean time to detect (MTTD) and mean time to respond (MTTR). Having precise detections means your team can start with a high-confidence assessment, not a clue.

It's important to clarify: precision is a metric, and correlation is the activity leading to it. But it's also more than just a single number. A myopic focus on reducing false positives is dangerous; it can lead to over-tuning your systems and creating blind spots. A truly precise system must be measured across a quadrant of outcomes:

- **True positives** (hits): The real attacks you successfully catch.
- **False positives** (false alarms): The benign activity you incorrectly flag as malicious.
- **True negatives** (correctly ignored): The vast sea of normal activity you correctly ignore, which is a powerful measure of your baselining efficiency.
- **False negatives** (misses): The real attacks you miss. This is the most critical and often the most difficult metric to measure, but it's the one keeping security leaders up at night.

By focusing on all four sides of this quadrant, you can build a more mature and resilient program. A system with a low false-positive rate but a high false-negative rate is a dangerous one. It makes you feel secure while leaving you vulnerable. The goal is to maximize true positives and true negatives, while minimizing both types of errors.

Metrics that actually matter

It's important to analyze metrics vendors are promoting with a healthy skepticism. So how do you know which metrics are indicative of good detections versus being a vanity metric? No one cares how many rules you have or how many alerts you closed if you're still getting breached.

Metrics that matter:	Metrics that don't:
Signal-to-noise ratio: How many raw events produce one actionable detection? Lower is better.	**Number of alerts processed:** This just measures how much noise you're dealing with.
Mean time to triage: How long does it take to determine if an alert is real or not? Faster is better.	**Number of detection rules:** This measures your vendor's marketing department, not your security posture.
Detection coverage: What percentage of likely attack techniques (MITRE ATT&CK) can you reliably detect?	**Mean time to resolution:** This can be gamed by closing tickets quickly, regardless of whether a threat was actually resolved.
Analyst productivity: How much time do your analysts spend on meaningful threat hunting and remediation versus low-value triage? More high-value work is better.	**Number of events blocked:** This vanity metric includes every fat-fingered URL and routine security control activation.

Additional practical metrics to consider include:

- **Environmental applicability:** Does the detection hold us across different environments and customer contexts?
- **Modification frequency:** How often does the SOC request changes to a specific detection?
- **Analyst queue behavior:** How quickly do analysts open specific alert types versus skip over them?

A measurement-first approach prevents the common mistake of trying to solve everything at once without understanding what's actually broken. And the best, most straightforward measurement if you're stuck is: if I put this detection in front of a SOC analyst, how do they respond? If they're recoiling, it needs work, regardless of any quantitative measurement.

The bottom line

Your security stack isn't failing because you lack tools. It's failing because your tools aren't working together intelligently. The solution isn't more dashboards, more AI, or more analysts. It's better detection engineering and better partners who take a systematic approach to turning data into actionable intelligence.

This requires discipline, not a bigger budget. It requires thinking like an engineer, not a consumer. The good news is, you don't need to rip and replace everything. You just need to build better connections between what you already have, tune out the noise, and focus your team on the signals that actually matter. Because at the end of the day, security is about one thing: stopping threats before they become incidents. Everything else is just posturing.

About Expel

Expel is the transparency-first managed detection and response provider that gives customers the same real-time platform its analysts use — no black box, no hidden processes. The company responds to critical incidents in 17 minutes and integrates with 130+ existing security tools, eliminating the rip-and-replace approach that defines traditional MDR.

Learn more at Expel.com

10/22/25