ON FALSE POSITIVES, LEGITIMACY AND CONTINUOUS TRUST

The question is not “is this malicious”

Almost every alert a security platform raises describes behaviour that is genuinely unusual and genuinely authorised. Deciding between those two is a different problem from detecting the behaviour, and it needs its own machinery.

IN BRIEF

A detection engine answers whether something happened. It cannot, on its own, answer whether it was supposed to happen — and that second question is where the analyst's day goes. Treating legitimacy as a first-class judgement, made by a component with different inputs from the detector, changes what reaches a human and what a machine may act on unattended. This paper describes the inputs that judgement needs, why tuning thresholds is not a substitute for it, the two failure modes it introduces, and how to tell a legitimacy engine from a confidence score with a friendlier name.

1 · The alert that was right and still wrong

At 02:14 a service account authenticates from a country it has never authenticated from, enumerates a directory, and reads several thousand records. Every element of that description is accurate, and the detection that fired was correct in the narrow sense that the behaviour it described did occur.

It was also a scheduled migration, approved three weeks earlier, running from a newly provisioned region because the previous one was decommissioned on Friday. An analyst establishes this in eleven minutes by opening a change ticket, a chat thread and a cloud console. The eleven minutes are the cost, and they are paid again the next night, because nothing in the platform learned anything.

This is the ordinary shape of a false positive, and calling it a detection failure misdiagnoses it. The detector did its job. What was missing was **an answer to a question the detector was never asked**: was this authorised? The industry's usual response — raise the threshold, add an exception, suppress the rule for that account — trades the eleven minutes for a permanent hole, and the hole is undocumented by design because exceptions are written in a hurry at 02:20.

FIGURE 1 · TWO QUESTIONS, TWO ENGINES

DETECTION

Did this happen?

Inputs: telemetry, baselines, signatures, models. Output: a described behaviour, with the events that show it.

LEGITIMACY

Was it supposed to?

Inputs: change records, entitlements, ownership, schedules, prior adjudications. Output: a verdict with the records it rests on.

Collapsing these into one score is the common design, and it is why tuning feels like a trade between noise and blindness. They are not the same question, they do not read the same data, and one of them has an authoritative answer sitting in a system nobody connected.

2 · What legitimacy is made of

The useful property of the legitimacy question is that it is frequently *answerable from records that already exist* — they are simply held outside the security platform, in systems that were never treated as security telemetry.

Intent declared in advance. Change tickets, deployment pipelines, maintenance windows, migration plans. An action inside an approved window, performed by the identity named in the approval, is a different object from the same action at an arbitrary hour.

Entitlement, as granted rather than as used. Whether an identity *may* do a thing is a fact about the directory. Whether it ever *has* is a fact about the estate. The interesting cases are where those disagree — the permission nobody exercised for a year, exercised tonight.

Ownership and purpose of the asset. A build agent reading source is expected; a finance workstation reading the same repository is not. Most estates hold this knowledge in a spreadsheet, which is to say they hold it nowhere a machine can consult at 02:14.

Prior adjudication. The single most valuable input, and the one most often discarded: an analyst decided this exact pattern was legitimate, gave a reason, and that decision is retrievable and reusable — with an expiry, because a reason that was true in March may not survive the reorganisation in June.

Two things follow. First, legitimacy is not derived from telemetry, so a platform that ingests only telemetry cannot compute it — the change system and the directory have to be event sources like any other. Second, because these inputs are records rather than inferences, **a legitimacy verdict can cite them**. “Authorised” is

not a judgement to be taken on trust; it is a claim with a ticket number attached.

> **The asymmetry that makes this work.** An adversary can imitate behaviour arbitrarily well — that is what living-off-the-land means. Imitating the record of prior authorisation is much harder, because it requires compromising a system whose whole purpose is to be an independent account of what was agreed. Legitimacy is expensive to forge in a way that behaviour is not.

3 · Why tuning is not the same thing

Every platform offers threshold tuning, allow-lists and suppression rules, and they are genuinely useful. They are also structurally different from adjudicating legitimacy, in three ways worth being precise about.

3.1 · An exception is unconditional; a verdict is conditional

Suppressing a rule for a service account silences it at 02:14 during the migration and equally at 14:00 in November when the account is being used by someone who should not have it. A legitimacy verdict is tied to the conditions that justified it — this window, this approval, this source — and stops applying when they stop holding, without anyone having to remember to remove it.

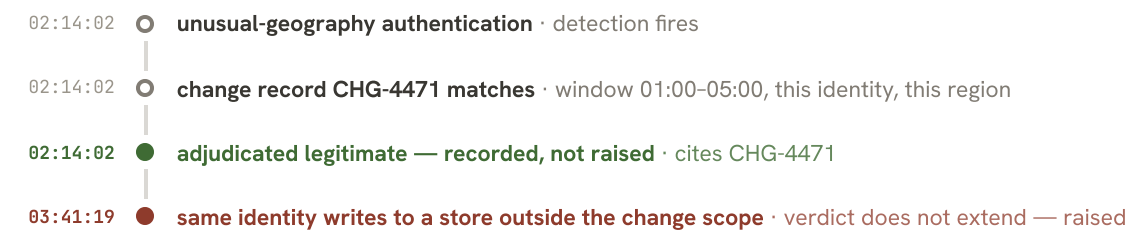
3.2 · Tuning destroys the signal; adjudication reclassifies it

A suppressed detection produces no record. An adjudicated one produces a record that says the behaviour occurred and was found authorised, on this basis, by this person or rule. The first leaves an estate whose quiet is uninterpretable; the second leaves a corpus — and that corpus is what makes “this identity's authorised-anomaly rate tripled this quarter” a question anyone can ask.

3.3 · Thresholds move the boundary; they do not sharpen it

Raising a threshold trades false positives for false negatives along a fixed curve; the only way to improve the curve itself is to add information the detector does not have. Legitimacy inputs are exactly that information — which is why a platform can be simultaneously quieter and more sensitive after connecting a change system, an outcome tuning cannot produce at any setting.

FIGURE 2 · THE SAME NIGHT, ADJUDICATED



Suppression would have covered 03:41 as readily as 02:14. The verdict covers only what the approval described — and the escalation is stronger for it, because an identity operating *outside* a window it is otherwise operating inside is a sharper signal than the anomaly alone.

4 · The two ways this goes wrong

A legitimacy engine is not free, and a paper that did not say where it fails would not be worth reading.

4.1 · The authorisation record is itself a target

If “matches an approved change” closes an alert, then creating an approved change becomes an attack step — and in most organisations the change system has weaker controls than the security platform consuming it. The mitigation is not to distrust the record but to treat *its creation* as an event in the same graph: who raised CHG-4471, when, from where, and did that identity's own behaviour look ordinary. An adversary who has to forge both the behaviour and its authorisation, and survive the correlation between them, is doing markedly more work than one who has to forge neither.

4.2 · Stale legitimacy is worse than none

A verdict with no expiry becomes an allow-list with better manners. Every adjudication needs a lifetime, a stated basis, and a review when the basis changes — a role reassignment, a decommissioned system, a closed ticket. The honest implementation makes the population of live verdicts visible and countable, because a legitimacy corpus that only grows is one nobody is reading.

> **The unifying idea.** Legitimacy is a claim about authorisation, made from records, carrying its own citation and its own expiry. Anything that cannot be cited or cannot expire is a suppression rule, whatever the interface calls it.

5 · What it changes for the operator

The queue changes composition, not just length. What remains is disproportionately the genuinely ambiguous — which is harder work per case and better work overall, and it should be said plainly that this is not a promise of an easier shift.

Unattended action becomes defensible. An edge may act on its own when the case is both behaviourally clear and legitimately unsupported. That conjunction, not a confidence percentage, is what a security leader can actually sign off on.

Analyst decisions stop evaporating. The eleven minutes are spent once and become an asset with a citation, rather than a closed ticket nobody will find at 02:14 tomorrow.

"Authorised" becomes measurable. How much of your estate's unusual activity is explained by approved change is a real number, and it is a better health metric than alert volume — which mostly measures how the thresholds were set.

6 · Questions worth asking, whoever you are buying from

01 When an alert is closed as authorised, what does the platform retain — a status, or a citation?

02 Can a verdict be scoped to a window, an identity and a resource, so it stops applying outside them?

03 Does every adjudication have an expiry, and can I see how many live ones exist right now?

04 Is the change system a first-class event source, or an integration that writes a comment on a ticket?

05 If an approval record is forged, what in your design notices? Show me the correlation, not the policy.

06 What is the difference in your product between suppressing a rule and adjudicating an instance?

07 Which automated actions require legitimate-support-absent, and which fire on detection confidence alone?

08 Can you show me the proportion of anomalies your platform explained by approved change last month?

7 · Where this sits in the series

Paper 01 argued that the sensor and the enforcement point must be one process if the gap between knowing and acting is to close. Paper 02 argued that the alert must carry its own evidence if anyone is to authorise action on it. Both leave a question open, and it is the one that decides whether unattended enforcement is responsible or reckless: *on what basis does the platform conclude that nobody meant to do this?*

Answering it with a threshold is how automated response earned its reputation. Answering it with records — cited, scoped and expiring — is what makes the answer reviewable months later by someone who was not there. The loop the first paper described only closes safely if this judgement is a component rather than a setting.

> *Unusual is a property of behaviour. Unauthorised is a property of the record. A platform that cannot tell them apart will keep asking people to.*



EventLake

ABOUT THIS PAPER

Paper 03 in a series on the architecture of active defense, published by **EventLake Research**. Paper 01, *The gap between knowing and acting*, and Paper 02, *An alert is a subgraph, not a row*, are its companions. The series is deliberately vendor-neutral, and the questions in §6 are meant to be asked of us as readily as of anyone else.

EventLake treats legitimacy as a distinct engine with its own inputs: change records, entitlements, asset ownership and prior adjudications, each verdict scoped, cited and given an expiry. **eventlake.com**