SOLUTION DESCRIPTION

Unified SIEM + UEBA on a sealed lake

Correlation and behavioural analytics over one substrate, where every finding can be traced back to the sealed events that produced it.

The problem with two products

SIEM and UEBA are usually bought separately and joined by an integration. The SIEM holds the events; the UEBA holds the baselines. Each has its own copy of your identity data, its own idea of what a host is, and its own retention window. The integration between them moves alerts, not evidence.

The consequence shows up in the one question that matters during an investigation: **why did this fire?** The UEBA says an identity deviated from its baseline. The SIEM holds the events that would show whether that deviation was interesting. Reuniting them is manual, and the answer is only as good as the analyst's ability to reconstruct a query that the UEBA never wrote down.

The second consequence is quieter and worse. Baselines computed in one system and events retained in another **drift apart on different schedules**. When the SIEM's ninety-day window rolls, the UEBA's baseline still references what was in it. Nothing errors. The behavioural model simply becomes an assertion about data that no longer exists.

One substrate, four consequences

A baseline is a query over the same events

Behavioural models are computed from the sealed lake rather than from a private copy. When a finding says an identity deviated, the events that establish both the baseline and the deviation are addressable — and they are the same records the correlation rules saw.

A finding carries its subgraph, not a query

The events that justified a conclusion are stored with it. Opening a finding two years later shows what the system actually saw, not what a re-run returns today against a window that has since rolled.

Rules and models see the same identity

Subject resolution happens once, at ingestion. A correlation rule and a behavioural model referring to the same person or workload are referring to the same node — which is what makes a cross-domain path one investigation rather than three.

Retention is one decision

Declared per record class, with a reason, and it governs everything computed from that class. A baseline cannot outlive the events behind it, because there is no second store for it to survive in.

What "sealed" means here, precisely. Events are append-only and carry per-record integrity proofs against an independent time source. It does not mean the lake is immutable in the sense of un-erasable — an erasure request destroys a subject's key, which renders their records unreadable without rewriting history. Sealing and erasure are compatible because one governs *alteration* and the other governs *readability*.

What this changes in an investigation

An analyst opens a finding and sees the path: the authentication, the process that read a credential store, the role assumed, the resource reached. Each hop is an event with its own provenance; the behavioural judgement that raised the finding is one node in that path rather than a separate system's verdict pasted on top.

The practical gain is not speed of triage — it is that **the analyst's own decision becomes part of the record**. Adjudicating a finding as legitimate writes a scoped, expiring verdict citing the basis, which is available to every subsequent evaluation. In a two-product estate that decision lives in whichever console the analyst happened to close.

What we will not claim

Unifying the substrate does not improve the detections themselves. A behavioural model that was weak in a separate product is the same model here — it simply becomes auditable. Anyone selling consolidation as a detection-quality improvement is describing a benefit that does not follow from the architecture.

It also concentrates risk. One store holding the correlation, the baselines and the evidence is one store to compromise. That trade is argued honestly in Paper 11, including the parts that do not favour us.

EventLake

NEXT STEP

Open a finding from a year ago in your current tooling and ask whether its evidence is stored or re-queried. That single test separates the two architectures better than any comparison table.

Further reading: Paper 02 *An alert is a subgraph, not a row* · Paper 03 *The question is not "is this malicious"* · Paper 06 *Evidence should be a query*. All published, none behind a form. **eventlakes.com**