SOLUTION DESCRIPTION

Cloud-native and Kubernetes estates

Where the subject of an alert may not exist by the time anyone reads it, and most of the identities are not people.

What is different about this estate

Three assumptions that hold on a server estate stop holding here. **Subjects are ephemeral** — a pod that lived four minutes is the subject of an alert that arrives at minute six, and "isolate the host" names something that has already gone. **Identity is overwhelmingly non-human** — workload identities, service accounts and pipeline credentials outnumber staff by an order of magnitude. And **the control plane is the estate**: an adversary with the right API permission does not need a shell anywhere.

A platform that treats a container as a small server will produce alerts about entities that no longer exist, attached to identities it cannot resolve, missing the layer where the consequential actions actually happen.

Four things the platform does about it

The workload, not the container, is the subject

Events resolve to the deployment and the image, so a behavioural baseline survives the replicas being replaced. A pod that has exited is still a queryable path, and the alert names something an engineer can act on.

Control-plane audit is telemetry, not a side channel

API calls, role bindings, admission decisions and secret reads are events in the same graph as process and network activity — which is the only way the sequence "credential read, then role assumed, then workload created" is one path rather than three tools.

Non-human identities are inventoried and baselined

Each is inventoried as an identity rather than as a secret. Workload behaviour is narrow and stable, so a service account doing something new is a sharper signal than a person doing something new — and almost nobody alerts on it.

Response verbs that fit the runtime

Suspending a workload identity, revoking a token, or blocking egress from a namespace are reversible and delegable. Deleting a workload or rotating a shared credential is not, and holds for a person's approval — with the six dependent services named.

On blast radius in a scheduler. Blocking egress from a namespace can sever more than it appears to, and the dependency graph changes on every deploy. Scope is therefore computed against the estate as it is at fire time — never from a description written when the playbook was authored, which in this environment is stale within a sprint.

The pipeline is part of the estate

Build systems hold long-lived credentials to production and run code authored by anyone with commit access. They are the highest-value, least-monitored surface in most cloud-native organisations. Treating build hosts and their identities as first-class subjects — with baselines, entitlement measurement and session re-evaluation — closes the path that a container-only view cannot see.

What we will not claim

A managed control plane exposes what its provider exposes. Where audit coverage has gaps, those gaps are recorded as collection gaps rather than papered over — because “we looked and found nothing” and “we were not collecting that” are different answers and only one is exculpatory.

We also do not see below the node. Firmware and hypervisor-level compromise is outside what a workload sensor observes; the compensating controls are attestation and provider assurance, and you should require them from someone.

EventLake

NEXT STEP

Start with the count: how many non-human identities exist in your estate, and which of their granted entitlements have never been exercised? If answering takes a project, that is the finding.

Further reading: Paper 07 *Authentication is a moment; trust is a duration* · Paper 09 *The integration is where the model gets decided.* All published, none behind a form. **eventlake.com**