

# How Onam finds the paths that matter

From read-only telemetry to a verified, MITRE-mapped, priced attack path — the method, end to end.

7 clouds, read-only

One property graph

5-domain verification

MITRE ATT&CK

Choke points

FAIR pricing

## THE ARGUMENT, IN ORDER

|           |                                                       |                                                                                                           |
|-----------|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>00</b> | <b>Executive summary</b>                              | A finding is a fact.                                                                                      |
| <b>01</b> | <b>The problem with a list</b>                        | A modern multi-cloud estate generates findings faster than any team can trace them.                       |
| <b>02</b> | <b>Ingestion — read-only, agentless, 7 clouds</b>     | Onam holds no standing credential in your cloud — only a reference it exchanges for a token that expires. |
| <b>03</b> | <b>One data model — the property graph</b>            | Everything normalises into a single per-tenant property graph.                                            |
| <b>04</b> | <b>Search — BFS from entry to crown jewel</b>         | Search is breadth-first from internet-reachable entry points toward crown jewels...                       |
| <b>05</b> | <b>Verification — five domains before "confirmed"</b> | A path that exists in the graph is a hypothesis, not a finding.                                           |
| <b>06</b> | <b>MITRE mapping, choke points, and price</b>         | Each verified hop maps to a MITRE ATT&CK technique...                                                     |
| <b>07</b> | <b>Summary</b>                                        | One claim, repeated at every stage: a finding is a fact, and only a verified route is a decision.         |

READS WITH [02 Cloud risk in dollars](#) · [03 One graph, one data model](#) · [04 Security & trust](#)

## 00 Executive summary

**A finding is a fact. A verified path is a decision.**

Every cloud scanner produces thousands of findings. The hard problem is not detection — it is knowing which handful of them combine into a route an attacker could actually walk to something that matters, and what that route is worth in dollars. Onam treats that as a graph-reasoning problem, solved the same way every time.

This paper describes the method end to end: how Onam ingests 7 clouds read-only, normalises everything into one property graph, derives the relationships an attacker would exploit, searches for routes from internet-reachable entry points to crown jewels, and — critically — verifies every edge across five independent security domains before it is ever called a confirmed path. Each hop is mapped to a MITRE ATT&CK technique; the full set of paths is reduced to its top five choke points; and the resulting exposure is priced with the FAIR model, so the output is a board-ready number rather than a severity label.

**The claim, precisely.** Onam does not promise to block attacks at runtime. It promises something more useful for prioritisation: a reproducible, verified and priced map of the routes that exist in your cloud right now — computed from read-only telemetry, with a deterministic identity for every finding so results are stable across rescans.

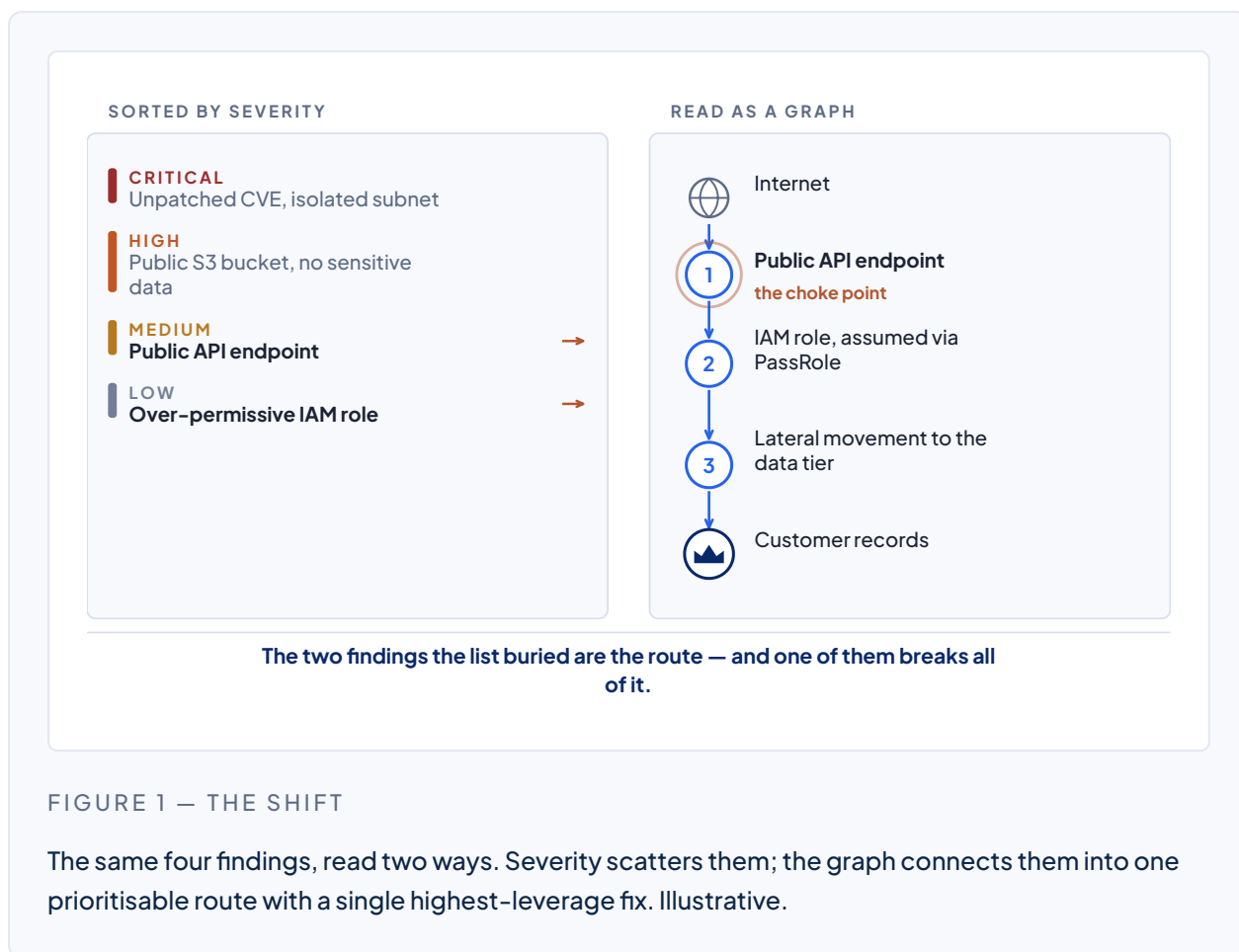
## 01 The problem with a list

A modern multi-cloud estate generates findings faster than any team can trace them. The industry's default answer is severity: sort by Critical, work down the list. But severity is a property of a finding **in isolation**. It cannot tell you that a "medium" public endpoint, a "low" over-permissive role and an unencrypted data store are — together — a single four-hop route to your customer records.

Attackers do not read your list. They chain. A route is only as strong as its weakest link, and the link that matters is rarely the one with the scariest CVSS score. To prioritise like a defender who thinks like an attacker, you have to reason over the relationships between findings, not the findings themselves.

## 02 Ingestion — read-only, agentless, 7 clouds

**Onam holds no standing credential in your cloud — only a reference it exchanges for a token that expires.**



The method starts with telemetry that is safe to collect. Onam connects to each cloud with the lowest standing privilege that permits a security read, and stores no long-lived credentials — only a reference to the role or principal, exchanged for short-lived tokens at scan time.

| Cloud         | Connection                                                               | Privilege posture                                        |
|---------------|--------------------------------------------------------------------------|----------------------------------------------------------|
| AWS           | Cross-account IAM role via sts:AssumeRole, unique External ID per tenant | SecurityAudit + ReadOnlyAccess; STS credentials ≤ 60 min |
| Azure         | Entra service principal                                                  | Reader + read-only Graph                                 |
| GCP           | Service account or keyless Workload Identity Federation                  | 6 read-only roles                                        |
| OCI           | IAM user + signing key                                                   | inspect + read only                                      |
| Alibaba / IBM | RAM read-only · Service ID                                               | Read-only, exchanged for a short-lived Bearer token      |
| Kubernetes    | Read-only ServiceAccount                                                 | get / list / watch                                       |

Full detail of the trust model is in Whitepaper 04.

### 03 One data model — the property graph

Everything normalises into a single per-tenant property graph. Assets are nodes; edges are typed — containment, attachment, IAM trust and network reachability. Catalog-driven derives then add the *abuse* edges: the relationships an attacker would actually use, as distinct from the ones the cloud provider documents.

This is the step that makes cross-cloud reasoning possible at all. A path that starts in one provider and ends in another is only visible when both are described in the same vocabulary. Architecture detail is in Whitepaper 03.

### 04 Search — BFS from entry to crown jewel

Search is breadth-first from internet-reachable entry points toward crown jewels — the assets a customer has designated as mattering. Breadth-first matters: it finds the *shortest* route first, and the shortest route is usually the one an attacker takes.

## 05 Verification — five domains before "confirmed"

A path that exists in the graph is a hypothesis, not a finding. Before Onam calls a path confirmed, every edge is verified across five independent security domains — identity, network, data, workload and configuration state. An edge that only one domain supports does not survive.

**Why this gate exists.** An unverified path is worse than no path: it spends a security team's credibility on a route that may not exist. Verification is the difference between a graph that suggests and a graph that asserts.

## 06 MITRE mapping, choke points, and price

Each verified hop maps to a MITRE ATT&CK technique, so the route reads as an attack narrative rather than a list of resource IDs. The full set of paths then reduces to its **top five choke points** — the nodes appearing on the most confirmed routes, where one change removes the most attack surface.

Finally the exposure is priced with FAIR, producing an Annualised Loss Expectancy per path and a roll-up across the estate. Method detail is in Whitepaper 02.

## 07 Summary

**One claim, repeated at every stage: a finding is a fact, and only a verified route is a decision.**

- Severity ranks findings against each other; a graph ranks routes against reality.
- 7 clouds, read-only and agentless, into one property graph with typed abuse edges.
- Breadth-first search from reachable entry points to designated crown jewels.
- Five-domain verification before any path is called confirmed.
- MITRE mapping for narrative, choke points for leverage, FAIR for the number.
- Onam does not block at runtime. It tells you, reproducibly, which few fixes matter most.