Onam Security

Ten lines of rule for every line of code.

What it is actually like to build the engine underneath 9,853 posture rules.

Anup Yadav · 2026-08-07 · onamsecurity.com

9,853 posture rules exist without 9,853 code paths.

The platform loads and evaluates rules. It never hardcodes them.

- **27,682 YAML to 2,590 Python** a 10.7:1 ratio of declarative rule to implementation code
- **A new detection** a file, not a deployment. That is why the catalog can be this large
- **CSPM** cloud security posture management: the rules that say a configuration is unsafe

Source: git ls-files (threat-engine, 2026-08-07) · facts/product.yaml

You would not write rules. You would build the engine that evaluates them.

The rules are content. The engine, the graph, and the pricing model are the work.

- **Rule authoring** a catalog problem with its own pipeline, and not the engineering role
- **The engineering role** traversal, cross-cloud correlation, and pricing what the graph finds
- **The consequence** the interesting bugs live in the engine, worth fixing once

Source: fact-base.md §1

The shape of the system, measured.

The graph is the hard part, and it was rebuilt from scratch this year.

attack-path-v2 is the second-largest body of code in the repo.

- **288 tracked files** second only to vulnerability at 400
- **A path** not a list of findings, but a reachability question across identity and data
- **A choke point** the one node on the most paths, where one fix beats ten

Source: git ls-files engines/ (2026-08-07)

261ac6736

the commit that finished retiring attack-path v1 — phases 1 through 6, July 2026

engines/attack-path/ now contains nothing but build caches. The old thing is actually gone.

Source: git log (threat-engine)

A finding is a row. A path priced in dollars is a decision.

The last mile is not detection — it is making the output rankable against everything else a business funds.

- **FAIR** the open standard, published by The Open Group, for putting a dollar value on cyber risk
- **ALE** annualised loss expectancy: the figure a board already knows how to weigh
- **The difference** severity compares findings to each other, dollars compares them to a budget

Source: fact-base.md §1 · marketing/facts/product.yaml

A 10-to-1 data-to-code ratio makes data quality the engineering problem.

And it is not solved. This is the honest state of it.

- **Azure alone** 422 duplicate rows across 225 repeated rule_ids in the registry
- **Found 2026-08-07** by a lint step that did not exist the week before
- **The catch** a data defect ships exactly like a code defect, with none of the tooling

Source: catalog/rule/all_rule_ids.csv (2026-08-07)

No runtime agent. If you want to build eBPF enforcement, that work is not here.

The honest limit, stated the way we would want it stated to us.

- **Read-only and agentless by design** nothing is deployed on the workload
- **The category fact** Wiz, Orca, Cortex Cloud and Microsoft all ship runtime agents
- **A real boundary** on the problem space, not a positioning choice

Source: fact-base.md §4.1

Three things to take away.

1 **The platform is data** 10 lines of rule per line of code, deliberately

2 The engineering is the engine, the graph and the pricing model

3 **The unsolved problem is data quality at catalog scale** and we said so first

Source: fact-base.md §1 (git ls-files, 2026-08-07)

Find your true north.

If the interesting part of that was slide 8, we should talk.
onamsecurity.com

onamsecurity.com