ONAM SECURITY

TECHNICAL WHITEPAPER · 05

Compliance, mapped once

How Onam evaluates a control once and reports it against 78 frameworks — and connects every gap to a priced attack path.

One rule → many controls

Rules as code

78 frameworks

Custom frameworks

Gap → priced path

THE ARGUMENT, IN ORDER

00 Executive summary

Evaluate the control once. Report it against every framework.

Compliance work is mostly duplication: the same control, re-checked and re-evidenced for PCI, then SOC 2, then ISO, then a customer questionnaire. Onam removes the duplication by decoupling the check from the framework. Rules are evaluated once as code; a single finding maps to many controls across 78 frameworks at once.

This paper explains the mapping model, why rules-as-code makes framework coverage extend automatically, and how compliance and attack-path reasoning reinforce each other — so a control gap is not just a failed check but a node on a priced path.

> **An honest framing.** Framework count is a coverage feature, not a differentiator on its own — several platforms publish large numbers. Onam's distinct value is the architecture that produces the mapping (one rule → many controls, auto-extending) and the ability to connect a compliance gap to a priced attack path. This paper leads with the mechanism, not the number.

01 One finding, many controls

Frameworks disagree about wording far more than they disagree about controls.

A framework is, at bottom, a set of controls. Many controls across many frameworks ask for the same underlying thing — "encrypt data at rest", "restrict public access", "enforce MFA". Onam evaluates that underlying condition once and maps the result to every control it satisfies or violates.

02 Rules as code — why coverage extends itself

A new rule extends every framework it maps to, and a new framework inherits every rule already written.

Onam's rules are versioned YAML the platform loads and evaluates; nothing is hardcoded. This is what makes framework coverage compound rather than accumulate by hand. When a new rule is added, it automatically extends every framework whose controls it maps to. When a new framework is onboarded, its controls map to existing rule IDs — no new checks to write.

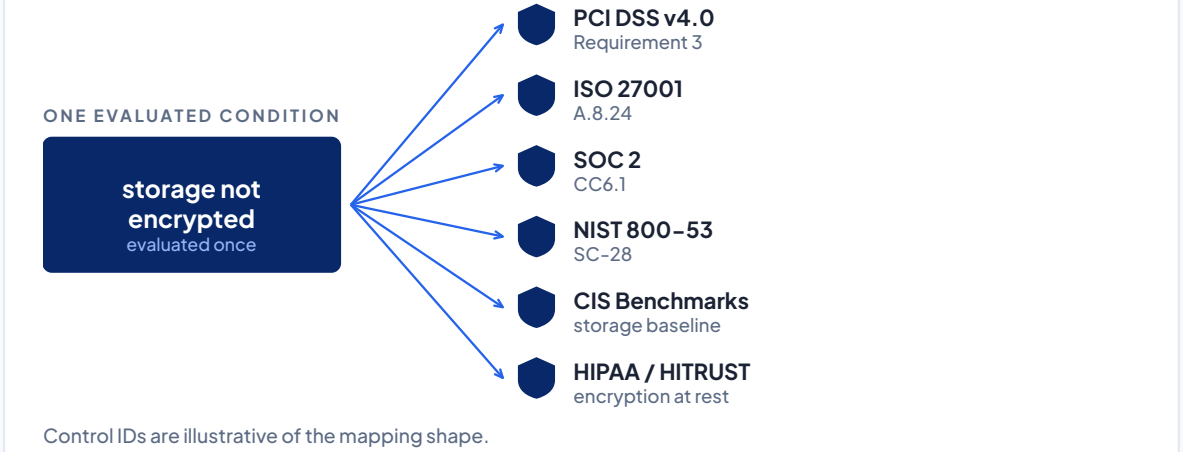

FIGURE 1 — THE MAPPING FAN-OUT

The duplication is in the reporting, not the work. Control IDs are illustrative of the mapping shape.

1. **Add a rule.** A new YAML rule ships; every mapped framework gains coverage for the condition it checks.
2. **Add a framework.** A new framework's controls map to existing rule IDs; coverage appears without new detection code.
3. **Add a custom framework.** A Rule Builder and custom-framework mapping let a team model an internal standard against the same rule library.

> **The compounding effect.** Because checks and frameworks are decoupled, work done for one obligation pays down others automatically. A rule written to satisfy an internal standard may also close a PCI and an ISO control the same day — without anyone re-authoring the check.

03 Frameworks supported

Coverage here means continuous evaluation, not an audit export generated the week before.

Onam supports 78 frameworks out of the box across regulatory, industry and best-practice standards — for example PCI DSS v4.0, SOC 2, ISO 27001/27017, HIPAA/HITRUST, GDPR, NIST 800-53 and CIS benchmarks — with custom frameworks mappable to rule IDs. Coverage is delivered as continuous evaluation, not a point-in-time audit export.

The mapping rests on a catalog of 11,433 rule definitions across 7 clouds and 549 cloud services, of which 9,853 are CSPM posture rules.

04 Compliance meets the attack path

Two control failures of equal audit weight are not equal risk.

A failed control is a compliance fact. The same failing condition is also a node in the property graph — which means Onam can tell you not just that a control failed, but whether its failure lies on a verified, priced route to a crown jewel. That reframes remediation: two "failures" of equal audit weight are not equal if one sits on a choke point and the other reaches nothing.

Lens	Question it answers	What it ranks by
Compliance	Which controls are failing, and where is the evidence?	Framework obligation
Risk	Which failure is on a reachable path to something that matters?	Priced exposure (FAIR/ALE)

Read together, the two lenses turn an audit backlog into an ordered queue: the control gaps that are also choke points come first, because fixing one of them closes an obligation *and* removes a route.

05 What framework count does and does not prove

A large framework number proves breadth of mapping. It does not prove depth of evaluation, quality of evidence, or that the controls most relevant to your obligations are covered well.

- **It does prove** that a platform has invested in mapping, and that onboarding a new obligation is unlikely to require new detection work.
- **It does not prove** that any individual control is checked the way your auditor expects, nor that evidence export matches your reporting format.
- **The question to ask any vendor**, including us: show me the rule behind this control, and the evidence it produces.

> **The honest limit.** Onam evaluates posture continuously and maps it to controls. It does not replace an auditor, and it does not produce a signed attestation. What it produces is the evidence an auditor asks for, kept current between audits rather than assembled the week before one.

06 Summary

Fix the condition once; every framework that maps to it updates itself.

- One evaluated condition maps to many controls across 78 frameworks — fix once, update everywhere.
- Rules-as-code means coverage extends automatically when either a rule or a framework is added.
- A control gap that sits on a priced attack path is not the same as one that reaches nothing, and Onam ranks accordingly.
- Framework count is a coverage feature. The architecture that produces the mapping is the differentiator.