Evidence Semantics and Scanner Orchestration

A public technical paper for the Salesforce Security Observatory

Created by Luca Pacini

Field	Value
Version	1.3
Publication date	1 August 2026
Implementation baseline	`1b32dfe0`
Documentation snapshot	`f834844b`
External verification status	Private and self-attested until the sanitised source repository is public
Document licence	CC BY 4.0 (see Publication and licensing)
Intended reference implementation licence	Apache License 2.0, on sanitised source release

> **Evidence basis notice**
>
> The implementation baseline `1b32dfe0` is the private snapshot against which the Source-established statements in this paper were checked. The documentation snapshot `f834844b` is the real private commit containing the REQ-072 documentation merge; the changes between the two snapshots are documentation and requirements only, with no production implementation files altered. Both identifiers describe private, self-attested evidence and neither is a public proof source. Every Source-established statement in this paper is therefore self-attested until the sanitised repository is published.

Abstract

Security tooling becomes unreliable when it treats four different questions as though they were one: did the assessment execute, was the source available, what value did the source return, and what conclusion may safely be drawn from that value?

The Salesforce Security Observatory separates those questions. A completed process does not prove that every source was available. An empty result does not prove that the relevant population does not exist. A non-zero result does not by itself establish severity or policy breach. A control mapping does not create a compliance decision.

This paper describes a Salesforce-native, read-only assessment model in which scanner families run through a bounded asynchronous plan, retain sanitised evidence inside the subscriber organisation, distinguish successful zero results from unavailable or incomplete evidence, and expose limitations as part of the result.

Current public support status

Area	Public statement
Architecture and evidence model	Source-established in the reviewed private baseline; self-attested until the sanitised repository is public.
Representative runtime behaviour	Expected or runtime-observed only where explicitly marked. No universal edition claim is made.
Partial-state rendering across cards, drill-downs and exports	Expected release-gate requirement; not claimed as release-validated in this paper.
Running-user sharing, CRUD and FLS behaviour	Expected release-gate requirement pending exact-release-candidate persona validation.
Environment compatibility	Not assessed unless a specific validated environment is explicitly named.

> **Core proposition**
>
> Nothing found and nothing assessed are not the same result.

1. The problem: security evidence is not a single value

A conventional dashboard often reduces a complex assessment to a count, badge or score. That compression is useful only when the path from source to interpretation remains trustworthy.

A displayed value of zero can mean that a source was successfully queried and returned no matching records. It can also mean that the source was unavailable, permissions were insufficient, a credential was not configured, a query failed, only part of a composite source was assessed, the scanner did not run, or evidence was not retained at the selected level.

- Only a successfully assessed and sufficiently complete source supports a genuine zero statement.
- Unavailable evidence must not be converted into zero.
- Incomplete evidence must not be presented as complete.
- Framework mappings support review; they do not decide compliance.

> **Integrity rule**
>
> Execution outcome, source availability, metric value and permitted interpretation are independent dimensions. A security observation is valid only when the relationship between them is explicit.

2. The four-dimensional evidence model

2.1 Dimensions

The values below are the abstract model. They are not display tokens; the rendered product vocabulary is given in 2.2.

Dimension	Question	Typical values
Execution outcome	Did the scanner or orchestration step reach a known terminal state?	Completed; completed with warnings; failed; not reached; not selected
Source availability	Was the required source accessible and sufficiently complete?	Available; partial; unavailable; unsupported; not configured; not assessed
Metric value	What did the successfully assessed source return?	Non-zero; zero; category; date; bounded detail; no applicable value
Permitted interpretation	What statement is justified by the other dimensions?	Evidence observed; zero observed; partial or incomplete; unavailable; not assessed

2.2 Abstract model and rendered product vocabulary

The evidence states in this paper are an abstract model. The product surfaces use the canonical rendered vocabulary below. Every concrete evidence condition maps to exactly one rendered state. Related abstract categories may resolve to different canonical states according to the limiting dimension. The mapping is made in the mapping layer rather than at render time. "Partial" is a completeness qualifier, not a separate product state token.

Abstract state	Rendered vocabulary	Interpretation
Evidence observed	Contextual metric or finding label	A non-zero observation is shown with its source and limitation; no generic success token is required.
Zero observed	None found	The assessed source completed and no matching records were observed.
Partial or incomplete	Count plus visible Partial qualifier and an adjacent Not assessed or Unavailable reason for the missing portion	A partial count must never look complete.
Scanner issue or failed assessment	Unavailable, with the safe cause appended - for example, "Unavailable - prerequisite missing"	Must include a safe boundary and administrator next action.
Not run or not retained	Not assessed; Not assessed at this evidence level; or Not captured	Selected according to whether execution, evidence depth or retention is the limiting factor.
Outside applicable boundary	Not applicable	Used only when the question does not apply to the organisation or feature.

Review status such as "requiring review" is a property of a finding, not a state in this model. A finding is rendered with its own label and scope; it does not introduce an additional evidence state.

2.3 Genuine zero versus unavailable

A genuine zero requires the intended scanner family to run, the required source to be accessible, the operation to complete successfully, the source set to be complete enough for the question, no matching records to be returned, and retained evidence to distinguish that result from missing output.

Preferred wording: *No matching exposure evidence was observed within the assessed sources.* Avoid absolute claims such as *No exposure exists.*

---

3. The public scanner contract

Condition	Required behaviour	Prohibited substitution
Source acquisition succeeds	Process records within the defined scope and limits.	Do not claim sources were assessed when they were not queried.
Successful empty result	Render None found with the assessed source and limitation.	Do not convert a source-specific zero into an organisation-wide guarantee.
Source acquisition fails	Render Unavailable with a safe cause and next action.	Do not write a synthetic zero.
Part of a composite source succeeds	Preserve usable evidence, mark the count Partial, and identify the missing category safely.	Do not calculate or render a complete total.
Individual scanner exception	Isolate the family where safe, retain a non-green family state, and continue independent later work.	Do not erase the failure or relabel it as empty.
Orchestration transaction failure	Fail the assessment boundary, retain the safest available status, and stop unstarted work.	Do not present a partially executed plan as complete.
Known safe failure category	Persist an approved bounded reason code and safe next action.	Do not persist raw exceptions, stack traces or queries.
Unknown failure category	Fail closed with a generic cause and an administrator next action.	Do not invent precision.

Expected release-gate requirement: the same contract must be enforced across cards, drill-downs, comparisons, exports and supporting framework views. This paper does not claim that every surface is already release-validated.

---

4. Evidence confidence is a ladder, not a binary claim

Confidence level	Meaning
Source-established	Visible in the reviewed private implementation source. In this paper it remains self-attested until public source is available.
Test-observed	Exercised by an automated test under a controlled fixture.
Runtime-observed	Occurred in a reviewed organisation execution.
Release-validated	Passed the defined release gates on the exact release candidate.
Supported	Part of the approved product contract for a stated environment and prerequisites.
Expected	Architecture or platform evidence suggests the behaviour, but required validation is incomplete.
Not assessed	No reliable conclusion is available.

A single successful execution is runtime evidence for that execution. It is not universal proof for every organisation of the same edition.

4.1 Evidence-basis register for this paper

This register identifies the principal load-bearing implementation and release-contract claims made by this paper. It is not exhaustive.

Claim	Evidence basis
Salesforce-native execution and Observatory-owned evidence storage	Source-established (private baseline; self-attested)
No scan evidence leaves the subscriber organisation	Source-established design boundary (private baseline; self-attested)
Apex-owned canonical scanner plan	Source-established (private baseline; self-attested)
Bounded asynchronous segmentation	Source-established architecture; representative runtime support not universally claimed
Visible partial marker beside every partial count, drill-down and export	Expected release-gate requirement; not release-validated
Safe cause, boundary and next action on every non-green state	Expected release-gate requirement; not release-validated across every surface
Running-user sharing, CRUD and FLS enforcement	Expected release-gate requirement pending persona validation
Exclusion of raw IP values from retained and exported evidence	Expected release-gate requirement pending data-shape tests
Formula-injection-safe CSV export	Expected release-gate requirement; not release-validated

Claim	Evidence basis
Like-for-like comparison of retained evidence	Expected release-gate requirement; not release-validated
Edition-wide compatibility	Not assessed

5. Compatibility vocabulary

Term	Meaning
Validated	Tested successfully on the exact stated release candidate and representative environment.
Supported with prerequisite	Supported when the stated Salesforce feature, permission or subscriber-controlled credential setup is present.
Expected but not validated	The design appears compatible, but representative runtime proof is incomplete.
Unsupported	Outside the approved product boundary or known not to meet a required dependency.
Not assessed	No support conclusion has been reached.

No unpublished compatibility matrix is treated as a normative source. Where representative validation is incomplete, the public status is Not assessed or Expected but not validated.

6. Salesforce-native architecture

```
Authorised operator or schedule
  |
  v
Lightning Web Components dashboard
  |
  v
Apex controller and scan service
  |
  v
Apex-owned scanner plan
  |
  +--> Salesforce record and setup sources
  +--> Subscriber-authenticated self-callout for approved Tooling API sources
  |
  v
Sanitised Observatory-owned evidence records
  |
  v
Dashboard, safe drill-down, comparison and CSV export
```

6.1 Architectural properties

Property	Evidence-basis statement
Salesforce-native execution	Source-established: dashboard, orchestration, scanners and evidence storage run on Salesforce.
Read-only security assessment	Source-established boundary: scanners inspect data, setup and metadata sources and do not remediate security configuration.
Observatory-owned writes	Source-established boundary: writes are limited to the product's own scan, metric, finding, detail, asset and comparison records.
No evidence egress	Source-established boundary: no scan evidence leaves the subscriber organisation. The self-callout targets Salesforce APIs in that same subscriber organisation.
Subscriber-controlled authentication	Source-established design: approved Tooling API access uses a subscriber-configured credential and principal.
Sanitised evidence	Source-established design boundary; exact data-shape release validation remains a release gate.

6.2 Running-user visibility, sharing, CRUD and FLS

Evidence basis: Expected release-gate requirement.

User-facing queries and evidence access must enforce the running user's sharing, object permissions and field-level security through explicit checks and/or user-mode data access.

Any narrow system-mode lifecycle operation must be separately justified, inaccessible to ordinary personas and validated on the exact release candidate. This paper does not claim release validation until non-System-Administrator persona tests pass.

---

7. Why the scanner plan is segmented

Salesforce is a multitenant platform with per-transaction limits. A security assessment that combines data queries, metadata reads, callouts, evidence persistence and comparison state should not assume that every operation belongs in one transaction.

Source-established architecture: the canonical scanner plan is owned in Apex, not duplicated in the browser. Scanner families are assigned to bounded asynchronous segments according to callout and non-callout characteristics. The paper deliberately does not publish a fixed transaction count because that count is an implementation detail that may change during safe refactoring.

- Callout boundaries are explicit.
- Compatible non-callout work may share a transaction.
- Each segment starts in a fresh asynchronous context.
- A segment may enqueue the next bounded segment.

- Later work receives only the retained orchestration context it needs.
- Finalisation occurs only after the plan reaches its terminal boundary.

7.1 Why not one transaction?

A single transaction creates unnecessary coupling between unrelated sources and increases the risk that one limit or callout boundary prevents the remainder of the assessment from running.

7.2 Why not one job per scanner?

A separate transaction for every scanner family creates avoidable asynchronous overhead and a deeper chain. Bounded segmentation groups compatible work while isolating callout boundaries.

7.3 Why not let the browser choose the scanner list?

Client-owned scanner selection risks divergence between manual scans, scheduled scans and later API entry points. An Apex-owned plan keeps the ordered assessment contract consistent.

---

8. Failure isolation and finalisation

8.1 Individual scanner-family failure

Expected contract: where safe, an individual family failure is isolated. The failed family remains non-green, independent later families may continue, and the overall scan can complete with warnings. A completed-with-warnings scan is not equivalent to a clean or fully assessed scan.

8.2 Orchestration transaction failure

Expected contract: a transaction-level failure that prevents the planned chain from reaching its valid terminal boundary fails the assessment. Unstarted work remains unassessed and must not be presented as completed.

8.3 Finalisation boundary - Expected contract

Finalisation is permitted only when the orchestration plan reaches its terminal boundary. Final status must preserve warnings, unavailable families, retained evidence level and safe failure context.

> **Fail-closed does not mean silent**
>
> Every unavailable or unknown condition must carry a safe cause, its evidence boundary and an administrator next action. This is an Expected release-gate requirement, not a release-validated claim across every surface.

---

9. Source availability is part of the evidence

Availability state	Meaning	Rendering rule
Available and complete	The source was accessed successfully and was complete enough for the stated question.	Metric can be interpreted within its stated scope.
Partially available	Independent source components were assessed, but at least one required component was unavailable.	Preserve usable evidence; mark the count Partial; identify missing scope.
Unavailable	The required source could not be assessed.	Render Unavailable with a safe cause and next action.
Not configured	A subscriber-controlled prerequisite was absent or unauthorised.	Render Unavailable with the safe cause "prerequisite missing" and the documented next action. The state is assigned in the mapping layer, not selected at render time.
Not assessed at this evidence level	The requested detail was outside the retained evidence level.	Do not infer zero or resolution.
Not captured	The historical record did not retain the requested evidence.	Do not substitute current live state.
Not applicable	The question does not apply to the organisation or enabled feature.	Do not score as pass.

Expected partial-count rule: a partial count may be displayed only when a visible Partial qualifier is adjacent to the value and the unavailable portion is separately identified. A naked number is treated as false-complete and is not acceptable.

10. Safe reasons and diagnostic separation

10.1 Retained operational evidence

Safe cause	Retained wording boundary	Required next action
Prerequisite missing	Name the missing feature or credential category safely.	Configure or authorise the documented subscriber prerequisite.
Permission boundary	State that the source could not be assessed with the current authorisation.	Review the assessing persona and approved principal access.
Unsupported source	State that the source is outside the approved product boundary.	Use a supported environment or treat the question as Not assessed.
Platform or query failure	Use an approved generic category; do not retain the raw platform message.	Retry after checking prerequisites; escalate with transient diagnostics if repeated.
Unknown condition	Use a generic fail-closed cause.	Review prerequisites and rerun; escalate for controlled diagnosis if repeated.

10.2 Transient engineering diagnostics

Raw exceptions, stack traces, queries, HTTP headers, platform messages and sensitive identifiers are diagnostic material, not product evidence. They should remain transient, access-controlled and sanitised for controlled engineering diagnosis.

10.3 Normalisation invariants

- Equivalent safe causes normalise to the same bounded category.
- Unknown input fails closed to a generic category.
- Normalisation never emits raw input.
- Every retained non-green category has a safe next action.

---

11. Sanitisation and data minimisation

11.1 Data that may be retained

- Bounded counts, categories, dates and non-sensitive identifiers required for safe drill-down.
- Approved source names and evidence limitations.
- Sanitised findings, details and comparison markers.
- Approved reason categories and administrator next actions.

11.2 Data excluded by the design boundary

- Access tokens, refresh tokens and session identifiers.
- Client secrets, consumer secrets, passwords and private keys.
- Certificate bodies and sensitive raw headers.
- Raw exception messages, stack traces and raw queries.

Expected release-gate requirement: raw IP values must not be retained or exported. This paper does not claim release validation until data-shape tests confirm the boundary across Login History and related evidence paths.

11.3 Export safety

Expected release-gate requirement: exports must preserve source, completeness and limitation context, apply formula-injection-safe sanitisation, and avoid exposing values that are intentionally omitted from the dashboard.

---

12. The subscriber-controlled self-callout

Some approved metadata sources require a Salesforce callout to the subscriber organisation's own Tooling API. The authentication objects and principal are configured and authorised by the subscriber. No evidence is sent to an external evidence service.

- No hard-coded secret is part of the product source.
- Authentication remains subscriber-owned.
- The assessing persona receives only the approved principal access.
- Removing the approved principal assignment or disabling the credential stops the metadata callout path.
- Failure to authenticate is rendered as unavailable evidence, not zero.

Evidence basis: Source-established design boundary in the private baseline; self-attested until the sanitised repository is public. Operational support still depends on subscriber configuration and exact-release-candidate validation.

13. Historical evidence integrity

Historical review must interpret the selected scan using the evidence retained for that scan. It must not silently substitute current live state, current mappings or a different evidence level.

13.1 Comparison contract

Evidence basis: Expected release-gate requirement.

- Compare only like-for-like evidence levels and compatible source boundaries.
- Missing or incompatible evidence is unavailable, not resolved or improved.
- New, repeated, worsened and resolved findings require compatible retained evidence.
- Asset, licence and permission drift must preserve the baseline context.

13.2 Retention and evidence depth

Evidence basis: Expected release-gate requirement.

Retention may be bounded, but cleanup must preserve the latest successful scan and approved comparison baselines. Older noisy detail rows must be removed before evidence required for defensible comparison.

14. Supporting framework evidence is not compliance

The Observatory may map retained evidence to Security Benchmark for Salesforce control identifiers or to extended checks. That mapping helps a reviewer locate supporting evidence. It does not decide whether an organisation complies with a control.

Question	Observatory answer
Did the scanner family return usable evidence?	Scanner-family evidence state
What evidence applies to a finding?	Finding or detail evidence state
How does the evidence relate to a benchmark control?	Coverage or supporting-evidence state
Is the organisation compliant with the control?	Requires formal assessment outside the Observatory

15. Public wording discipline

Avoid	Use
No exposure exists.	No matching exposure evidence was observed within the assessed sources.
The organisation is secure.	The selected checks returned the evidence described in this assessment.
The scanner passed.	The scanner family returned usable evidence.
Clean.	None found within the stated assessed scope.
No users are affected.	No matching users were observed in the successfully assessed source.
The token is active.	OAuth grants requiring review were observed, with point-in-time activity evidence stated separately where available.
The control is compliant.	The retained evidence supports review of the mapped control identifier.
All Salesforce editions are supported.	Environment compatibility is Not assessed unless explicitly validated.
Platform limits are bypassed.	The scanner plan is segmented to execute within Salesforce transaction boundaries.

16. Validation philosophy

A defensible release process verifies more than code coverage.

- Source review of the canonical scanner plan and evidence-state contract.
- Automated tests for non-zero, zero, unavailable, partial and failure paths.
- Reason-code normalisation and raw-diagnostic rejection tests.
- Dashboard, drill-down, comparison and export contract tests.
- Controlled runtime execution in representative organisations.
- Package installation and upgrade evidence where distribution is claimed.
- Non-System-Administrator persona testing for sharing, CRUD and FLS.
- Public-copy sanitisation, link validation and overclaim review.

Coverage percentage is useful, but it does not prove that evidence semantics are correct. Tests must assert the distinction between zero, unavailable and partial evidence.

17. Known limitations

- The Observatory is advisory and read-only with respect to Salesforce security configuration.
- It does not remediate findings or prove that a retained OAuth authorisation is a currently usable token.
- It does not replay credentials or test access by attempting to use them.
- It does not provide real-time monitoring or replace formal audit activity.
- Some metadata evidence depends on a subscriber-authorised self-callout.
- Evidence visibility depends on approved user and principal permissions.
- A successful execution in one organisation does not establish universal edition support.
- Retention and evidence detail are intentionally bounded.
- Framework mapping provides supporting evidence, not certification.
- Unsupported or unavailable sources remain non-green and require review.

18. Design rationale summary

Decision	Rationale
Keep scanner-plan ownership in Apex	Prevents client, schedule and API entry points from drifting apart.
Use bounded asynchronous segmentation	Separates callout boundaries and resets transaction limits without publishing a brittle fixed job count.
Continue after isolated family failure	Preserves independent evidence while keeping the failed family non-green.

Decision	Rationale
Fail the scan on orchestration failure	Prevents a partially executed plan from appearing complete.
Treat source availability as evidence	Distinguishes genuine zero from missing visibility.
Persist approved reason categories only	Provides operational value without storing raw diagnostic material.
Keep detailed diagnostics transient	Reduces attacker-useful retained information.
Use subscriber-controlled credentials	Avoids hard-coded secrets and preserves subscriber ownership of authentication.
Retain evidence inside Salesforce	Keeps evidence within the subscriber platform and access model.
Compare compatible retained evidence only	Prevents omitted or incompatible evidence from appearing resolved or improved.
Keep benchmark mapping separate from compliance	Prevents supporting evidence from becoming an unsupported assurance claim.

19. Conclusion

A security dashboard is trustworthy only when it can explain the path from source to statement. The Observatory's proposition is not that it can produce more counts. It is that every count, zero, warning and unavailable state must retain enough context to explain what ran, what source was assessed, what was returned, what could not be assessed, what conclusion is permitted and what limitation or next action remains.

This approach treats evidence semantics as part of security engineering rather than presentation copy. The asynchronous plan, source-availability model, safe reason taxonomy, sanitised evidence boundary and historical comparison rules serve the same objective: prevent the system from claiming more than its evidence supports.

> **Closing principle**
>
> Nothing found and nothing assessed are not the same result.

References

1. Salesforce Developers, *Secure Apex Classes* - https://developer.salesforce.com/docs/platform/lwc/guide/apex-security
2. Salesforce Developers, *Call APIs from Apex* - https://developer.salesforce.com/docs/platform/lwc/guide/data-api-calls-apex.html

3. Salesforce Developers, *Get Started with Named Credentials* - https://developer.salesforce.com/docs/platform/named-credentials/guide/get-started.html
4. Salesforce Developers, *Named Credentials Glossary* - https://developer.salesforce.com/docs/platform/named-credentials/references/named-credentials-reference/nc-glossary.html
5. Salesforce Developers, *Package Named Credentials* - https://developer.salesforce.com/docs/platform/named-credentials/guide/nc-package-credentials.html
6. Salesforce Developers, *Populate External Credential Principals* - https://developer.salesforce.com/docs/platform/named-credentials/guide/nc-populate-external-credentials.html
7. Salesforce Developers, *Exploring a Combined Async Apex Framework* - https://developer.salesforce.com/blogs/2023/02/exploring-a-combined-async-apex-framework
8. Security Benchmark for Salesforce, project documentation - https://docs.securitybenchmark.org/
9. Creative Commons Attribution-ShareAlike 4.0 International - https://creativecommons.org/licenses/by-sa/4.0/
10. Creative Commons Attribution 4.0 International - https://creativecommons.org/licenses/by/4.0/
11. Apache License, Version 2.0 - https://www.apache.org/licenses/LICENSE-2.0

Each URL is printed in full so that references remain usable in printed and archived copies. Where the publication format supports hyperlinks, the printed URL is also the link target.

Publication and licensing

FortMilo is the public publication surface for the Salesforce Security Observatory. Luca Pacini is the creator of this paper and of the Observatory project.

The source reviewed for this version is maintained in a private pre-release repository. The reference implementation is intended for publication under the Apache License, Version 2.0 following sanitised source release; no public source-repository URL is claimed at this version. Until that release, Source-established claims in this paper remain self-attested against the private implementation baseline identified on the cover.

The CC BY 4.0 licence applies to FortMilo-authored text only. It does not apply to third-party trademarks or material accessed through the cited references. No Security Benchmark for Salesforce control text, audit procedure or remediation content is reproduced or adapted in this paper.