

READ-ONLY IS A PERMISSION, NOT A SCOPE CONCLUSION

A compliance integration can be unable to change one setting and still hold a privileged map of the identities, controls, vulnerabilities, and management plane surrounding the CDE. Read-only describes an operation. It does not finish the scope analysis.

PRACTITIONER POSITION

PCI DSS scope includes the CDE and system components, people, and processes that connect to it or could impact its security. My practitioner position is that a GRC, compliance-automation, or security-assurance integration should therefore be analyzed through its actual machine trust relationships-not through the platform category or a read-only label. The useful unit of analysis is Platform -> Integration -> Machine Identity -> Credential -> Effective Permission -> Target -> Information Observed or Action Allowed -> PCI Relationship -> Potential Compromise Consequence. A connector that reads unrelated training status may support an out-of-scope conclusion. A connector that enumerates CDE-relevant cloud IAM, network controls, vulnerability findings, administrative endpoints, deployment state, or security tooling presents a materially different proposition. Read-only remains relevant: it may limit direct modification capability. But the classification decision also needs to consider reconnaissance value, credential custody, consent grants, token handling, provider administration, accessible tenants and resources, alternate permissions, evidence sensitivity, and whether compromise could help affect the security of the CDE. The platform and each connector may reach different conclusions. Inventory and assess them separately.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

A compliance-automation platform displays a green cloud-configuration tile. The integration owner says the connector is out of scope because its service principal is read-only. The architecture inventory contains one row named GRC Tool. It does not name the OAuth application, credential store, consent grant, effective cloud roles, accessible subscriptions, CDE resource groups, configuration fields collected, vulnerability results synchronized, platform administrators, token revocation path, or logs proving use. During scoping, the team discovers that the connector can enumerate CDE assets, IAM trust relationships, security-group rules, logging gaps, public endpoints, and vulnerability findings. It still may not be able to change a resource. But the original conclusion was never a capability analysis. It was an adjective standing in for one.

WHAT SHOULD MAKE YOU STOP

- The inventory has one SaaS-platform record but no separate records for connectors, OAuth applications, service principals, API tokens, collectors, or locally installed agents.
- Read-only appears in the scope rationale without an effective-permission export, target inventory, collection schema, credential path, or compromise analysis.
- One platform-wide scope classification is inherited by HR, MDM, cloud, source-control, vulnerability, identity, and deployment integrations with materially different access.
- The connector observes CDE-relevant IAM, network, vulnerability, endpoint, logging, container, secrets metadata, deployment, or administrative information, but the data is labeled generic compliance telemetry.
- The entity knows which API was configured but cannot identify the machine identity, where its credential is stored, who can alter consent, how access is revoked, or which tenant administrators can retrieve the collected information.
- Documentation describes intended permissions while the assessment cannot reproduce effective permissions across every connected account, subscription, project, repository, or device population.
- The integration is treated only as an evidence source even though its failure, compromise, blind spot, or stale collection can alter the claims made about CDE security controls.
- No change trigger reopens the analysis when a connector, scope target, API permission, vendor feature, identity assignment, collection field, or CDE boundary changes.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Separate the nouns. Inventory the SaaS platform, each integration instance, collector or agent, OAuth application or service principal, credential or token, target account, permission grant, administrator population, collected dataset, evidence output, and dependent assessment claim as distinct but related records. The platform name is not a sufficient asset model.

Write effective capability, not marketing capability. Record the actions actually allowed, the resources actually reachable, conditions and inherited permissions, whether access is interactive or machine-only, token lifetime and refresh behavior, and what an administrator can reconfigure without redeploying the connector. Intended least privilege and observed least privilege are different evidence propositions.

Distinguish data-plane, management-plane, control-plane, and evidence-plane access. A connector may never encounter PAN while still reading the configuration of systems that protect payment flows, the identities that administer them, the vulnerabilities that expose them, or the telemetry used to conclude that controls operated. Absence of account-data access answers one question; it does not answer every security-impact question.

Analyze observation as a security capability. Ask what an attacker could learn from the collected inventory, topology, IAM relationships, network rules, findings, software versions, logging state, endpoint posture, repository metadata, and control exceptions. Reconnaissance alone does not create an automatic PCI

result, but it belongs in the fact-specific compromise and scope analysis rather than disappearing behind read-only.

Classify the platform and integrations independently. One connector may read general workforce training records, another may observe administrative workstations, and another may enumerate the cloud control plane containing the CDE. Preserve each rationale, effective period, evidence, reviewer, uncertainty, and change trigger. Do not average them into one vendor-level label.

Map third-party and customer responsibilities at the service-and-activity seam. Identify who secures the hosted platform, integration runtime, machine identity, credential store, consent and roles, target configuration, administrator access, logs, incident response, deletion, and evidence export. A provider security assertion does not document the customer's grants or implementation.

Make the trust path testable. Reproduce effective permissions; inspect connected targets; validate collected fields; review platform and target-side access logs; test revocation; identify administrators; confirm token storage and rotation; observe failure behavior; and trace one dashboard conclusion to the exact source population, collection time, method, exclusions, and exception path.

Let change propagate. A new CDE account, expanded permission, additional connector, changed collection schema, new platform administrator, vendor feature, modified retention setting, compromised token, or stale integration should reopen only the related scope, security, evidence, and assessment claims. Machine trust is architecture and should have the same change discipline as other architecture.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 Which exact platform, integration instance, machine identity, credential, target, and administrator population are being classified?
- 02 What effective actions and observations are possible across every connected account, tenant, subscription, project, repository, endpoint, or service?
- 03 Which CDE components or security-impacting systems can the integration observe, and what could the collected information reveal to a compromised platform or credential?
- 04 Where are credentials and refresh tokens stored, who can expand consent or permissions, and how are rotation, revocation, and emergency disablement proven?
- 05 Which provider activities and customer-controlled configurations support the trust path, and what evidence covers each responsibility?
- 06 Which assessment claims depend on the integration's output, and can each result preserve population, method, time, exceptions, limitations, and source lineage?
- 07 Why does each connector receive its own classification, and what unresolved fact would change that conclusion?
- 08 Which changes reopen the analysis, and can the entity detect them before the annual scope confirmation?

ESCAPE THE WACK

01 / FIND

Discover every GRC, compliance-automation, monitoring, CSPM, MDM, vulnerability, EDR, SIEM, CI/CD, identity, configuration, source-control, and assurance integration that authenticates to or observes the CDE or systems that could impact its security.

02 / MAP

Build an Integration Trust and Capability Ledger connecting platform, integration, machine identity, credential, effective permission, target, collected information, control plane, administrator, provider responsibility, evidence claim, and potential compromise consequence.

03 / EXPLAIN

Write a separate, dated classification rationale for the platform and each connector. State what read-only constrains, what it does not constrain, which facts support the conclusion, which uncertainty remains, and whose judgment the conclusion represents.

04 / REDUCE

Remove unused connectors, targets, fields, permissions, long-lived credentials, broad consent, excess administrators, duplicate evidence feeds, and indefinite retention. Prefer narrowly scoped identities and explicit revocation paths without pretending reduced capability eliminates the need for analysis.

05 / PROVE

Choose one integration that observes CDE-relevant controls. Reproduce effective access from both ends, trace one collected result to its source population and assessment claim, revoke the credential, and show that monitoring, ownership, and dependent-claim handling behave as documented.

THE LINE WORTH REMEMBERING

READ-ONLY LIMITS A VERB. IT DOES NOT CLASSIFY THE TRUST RELATIONSHIP.

Download the Integration Trust and Capability Ledger. Replace one row named GRC Tool with the connectors, machine identities, credentials, permissions, targets, observations, administrators, evidence claims, responsibilities, compromise consequences, and change triggers that actually exist.

START WITH THE SOURCE MATERIAL

These references ground the practitioner interpretation. They do not turn the metaphor or method into official PCI DSS terminology.

- WACK-SOURCE-0001 / PCI DSS v4.0.1 / PCI Security Standards Council / [Open source](#)

- WACK-SOURCE-0005 / PCI DSS Scoping and Segmentation Guidance for Modern Network Architectures / PCI Security Standards Council / [Open source](#)

AUTHORITY CLASSIFICATION / PCI_SSC_SOURCE / ASSESSOR_JUDGMENT / PRACTITIONER_POSITION / PRACTITIONER_METHOD / AI_ASSISTED

Practitioner reviewed by Scott Norton on 2026-09-09 against canonical digest

c2c90ced603304452e8d49035f408bbe54cab22a713c24eef0f012d203df4cdc. Educational content, not a compliance determination.

Independent practitioner resource; not affiliated with or endorsed by PCI SSC. Authorship and authority classification do not determine legal intellectual-property ownership.