PCI DSS 4.0 *Runtime Readiness* Checklist

Nineteen evidence checks across Requirement 6.3.2 and its neighbors, built from the questions QSAs are asking in 2026 assessments. Mark only what you can produce on demand today.

"Provide evidence that the 3,400 third-party components listed in your SBOM are evaluated for vulnerabilities relevant to your in-scope systems."

A representative QSA request from a recent PCI DSS 4.0 assessment. The SBOM answers the inventory. The checks below cover the three words doing the real work: **evaluated**, **relevant**, and **in scope**.

1 The declared inventory

Req 6.3.2

QSA ASKS *"Show me every component in your custom software, with provenance."*

- [ ] An SBOM exists for every in-scope application, regenerated on each build, in CycloneDX or SPDX
- [ ] Direct and transitive dependencies are listed with versions and hashes, attested back to source
- [ ] Custom and bespoke software is in the inventory, not only third-party components

2 The runtime layer

Req 6.3.2

QSA ASKS *"Which of these components loaded inside the CDE in the last 90 days?"*

- [ ] You can list the declared components actually observed in memory across in-scope services
- [ ] Every in-scope service is mapped to its CDE network zone and reconciled against the live deployment manifest
- [ ] Drift between the scope diagram and the running system is documented, with dates
- [ ] The declared vs. loaded delta is producible on demand for any service

3 Reachability and ranking

Req 6.3.1

QSA ASKS *"Is the vulnerable function on a code path your application actually executes?"*

- [ ] Open CVEs are classified reachable or not reachable from your code paths, not just present in the build
- [ ] Function-level evidence covers cryptographic libraries handling PAN and parsers on the payment-input path
- [ ] Your documented risk-ranking methodology cites runtime context: loaded, reachable, exposed

4 The justified backlog

Req 11.3.1.1

QSA ASKS *"Walk me through why these findings were deprioritized."*

- [ ] Every lower-severity finding left open carries a targeted risk analysis
- [ ] Justifications cite runtime evidence (not loaded, not reachable, not in the CDE), not assertions
- [ ] Each analysis is revisited on the cadence your policy defines

5 Remediation on a clock

Req 6.3.3

QSA ASKS *"Show me the one-month clock was met on critical patches."*

- [ ] Critical patches close within one month, with timestamps in the remediation log
- [ ] The remediation queue is keyed to the reachable-and-loaded subset, not the full SBOM
- [ ] A documented schedule covers everything below critical

6 Continuous operation

Reqs 11.3.1, 11.3.1.2

QSA ASKS *"Prove this program ran all year, not just the week before the assessment."*

- [ ] Internal vulnerability scans run authenticated, on the required cadence
- [ ] Runtime evidence is collected continuously, not assembled for the audit window
- [ ] Each quarterly review compares a loaded-in-the-CDE report against the SBOM

Reading your score

17 - 19	Audit-defensible. Your evidence connects the inventory to risk ranking and the remediation clock.
11 - 16	The inventory is in place. The runtime layer is what is missing, and it is where the questions now land.
0 - 10	An SBOM-only posture. Expect follow-up on "evaluated," "relevant," and "in scope."

10,000 declared. 37 loaded.

One representative engagement: of 10,000 declared dependencies, 37 were loaded in the CDE. The backlog built from the full list was almost entirely noise.

Kodem connects code and runtime, so the loaded, reachable subset is evidence you produce on demand rather than reconstruct for the audit. Runtime provides truth.

kodemsecurity.com