Skylos code-scan evidence mapped to SOC 2 Trust Services Criteria

Framework edition: 2017 TSC (revised points of focus 2022)

External summary: no code locations, finding text or exception justifications.

This document shows what Skylos, a code-scanning tool, checked in the code and build pipeline of Example Co (fictional sample), and what an auditor will still need from the company. Skylos Cloud generated it on 2026-11-10 09:00:00Z at the request of a workspace member.

It is not a SOC 2 report; only a CPA firm issues one.

What Skylos tested and what comes from your own systems

Criteria in scope	From your own systems	With a Skylos test	- tests passed	- gaps found
33	25	8	2	6

Of the 33 SOC 2 criteria in scope, Skylos has control tests that give evidence toward 8. It has none for the other 25: most of those are about people, access, vendors and infrastructure, and that evidence always comes from the company's own systems (HR, access management, vendor records), not from a code scan.

For the 8: the Skylos tests passed for 2 and found gaps for 6. A passed Skylos test covers only the part of a criterion it names, so every criterion also needs evidence that only you have.

Skylos scan-policy result (Skylos default security policy, revision 2). Not a SOC 2 Trust Services Criteria result or a determination of compliance:

Policy met, fixes due

Every project in scope had a trusted, complete Skylos scan in the last 90 days, and the open CRITICAL or HIGH findings are all still inside their fix-by dates. Fix them by those dates to keep this result.

2 projects in scope; 2 of 2 had a trusted, complete scan in the last 90 days with every check the policy needs. Scans of each repository's default branch, as resolved by Skylos Cloud at upload.

Do these first

1. Fix before the deadline: 3 open CRITICAL or HIGH findings, the first due 2026-11-14. After that date the policy will not pass.
2. Run a trusted scan of example-api's default branch by 2026-11-17: the code scanning control (SK-C1) expects one at least every 7 days.
3. 1 approved exception ends within 30 days (first 2026-12-10). Fix the finding or renew the exception before then.

16 more actions are in the full list of 19 under What to do next on the next page.

Check the record Skylos stored for this report (it is not signed), then compare its key facts with this copy:

https://skylos.dev/verify/report/00000000-0000-4000-8000-000000000002

How to read this report

- SK-C codes: The six controls Skylos tests from its own records (SK-C1 to SK-C6) and their tests, for example Scan before merge (SK-C1.1). Part A2 lists each one with its result.
- CC codes: The SOC 2 criteria as the Trust Services Criteria number them, for example Change management (CC8.1).
- DC codes: The nine items of the system description a SOC 2 report contains, for example System description item 3 (DC 3). Part A1 lists them.
- Deviation: An item in a test that did not meet the control, written "x of y" as a SOC 2 report writes it. It is a gap to fix or explain; an auditor decides what it means.
- Population: Every item a test looked at, for example every pull request merged in the period. Each population is a CSV file an auditor can pick samples from (part A5).
- Trusted scan: A scan uploaded from CI (GitHub or GitLab OIDC, or a project token an admin marked for CI) or run by Skylos Cloud. Uploads from a developer machine are unverified and never count.
- Skylos scan-policy result: Skylos's own rule about open CRITICAL and HIGH findings and their fix-by dates. It reads "Policy met", "Policy met, fixes due", "Policy not met" or "Not assessed". Section 3 has the rule.
- Parts and sections: Parts A0 to A5 are the SOC 2 part; A0 is laid out like section II of a SOC 2 report, A2 like section IV and A3 like section V. Sections 1 to 10 are the scan evidence behind them.

What to do next

To do: 3 to fix, 4 to change, 4 to write and 8 to collect.

Every gap this report found, in four groups. Each item says what to do and where to do it.

1. Fix now or before a deadline

Findings past or near their fix-by date, and the next scan your controls expect.

- Fix before the deadline: 3 open CRITICAL or HIGH findings, the first due 2026-11-14. After that date the policy will not pass. Criteria: CC4.2, CC7.1, CC7.2.
- Run a trusted scan of example-api's default branch by 2026-11-17: the code scanning control (SK-C1) expects one at least every 7 days. Criteria: CC4.1, CC5.2, CC6.8, CC7.1, CC7.2.
- 1 approved exception ends within 30 days (first 2026-12-10). Fix the finding or renew the exception before then.

2. Change a setting so Skylos can test more

Each change lets the next report test more of your controls.

- Run Skylos on every pull request before it merges, with the Skylos GitHub workflow or your own CI job, so each merge has a trusted scan (SK-C1.1). Criteria: CC4.1, CC5.2, CC6.8, CC7.1, CC7.2.
- Scan the default branch on every push and on a weekly schedule in CI, so no stretch without a trusted scan is longer than 7 days (SK-C1.2). Criteria: CC4.1, CC5.2, CC6.8, CC7.1, CC7.2.
- Require an approving review from someone other than the author in GitHub branch protection or a ruleset (SK-C3.3). Criteria: CC8.1.
- Add the Skylos check to the required status checks of the default branch in GitHub for example-web (SK-C3.4). Criteria: CC8.1.

3. Write what only you can write

Only management can write these. Skylos never writes them for you.

- Write the 6 empty system description items (DC 3, DC 4, DC 5, DC 7, DC 8 and DC 9) on SOC 2 answers, Description.
- Approve the wording of 3 Skylos-tested controls (SK-C4, SK-C5 and SK-C6) on SOC 2 answers, Controls.
- Record a corrective action for 3 deviations without a management response (SK-C1.1, SK-C1.2, SK-C3.3) on SOC 2 answers, Deviations.
- Adapt the draft of management's assertion in section A0 of the report and have an officer sign it before the audit.

4. Collect evidence Skylos cannot see

The auditor will ask for these. They come from your other systems, not from your code.

- Control environment (CC1.1 to CC1.5): Code of conduct and its acknowledgements, organization chart, management or board oversight, background checks, security training and performance reviews.

  Usually from: HR system, policy documents, a compliance platform (for example Vanta, Drata or Secureframe).
- Communication and information (CC2.1 to CC2.3): Security policies and how they are communicated internally; the system description and commitments communicated to customers; a channel to report security issues.

  Usually from: Policy documents, the internal wiki, customer contracts and the security page.
- Risk assessment (CC3.1 to CC3.4): An annual risk assessment, a risk register with owners and treatments, fraud risk, and how

significant changes are assessed.

Usually from: Risk register (spreadsheet or GRC tool), management meeting minutes.

- Monitoring and control activities (CC4.1, CC4.2, CC5.1 to CC5.3): Control activities chosen for the risks, and the policies and procedures that put them in place.

  Usually from: Policy documents, a compliance platform.

  - CC4.1: How the other controls are monitored: compliance-platform monitoring, internal audit or control self-assessments.

  - CC4.2: How control deficiencies reach management and are tracked to closure: tickets, meeting minutes.

  - CC5.2: Technology controls beyond code: access, infrastructure and operations policies and their operation.

- Logical and physical access (CC6.1 to CC6.8): User provisioning and removal records, periodic access reviews, MFA and SSO settings, firewall and WAF rules, physical security, asset disposal.

  Usually from: Identity provider (for example Okta or Google Workspace), cloud console, HR offboarding records; physical security usually from the cloud provider's SOC 2 report (carve-out).

  - CC6.1: MFA and unique IDs on critical systems, access provisioning and periodic access reviews, encryption key management: identity provider and cloud console settings.

  - CC6.7: Encryption in transit and at rest, endpoint and removable-media encryption: cloud console and device-management settings.

  - CC6.8: Endpoint anti-malware, operating-system patching and approved software: device management (MDM) and endpoint protection reports.

  - SBOM per release (SK-C6.2): a software bill of materials (SBOM) for each release; run skylos sbom in your release job or export one from your build system.

- System operations and incidents (CC7.1 to CC7.5): Security event triage, incident classification, response and recovery records.

  Usually from: Incident tickets, the on-call tool, post-incident reviews.

  - CC7.1: Infrastructure vulnerability scans, penetration test reports and production configuration monitoring.

  - CC7.2: Logging, alerting and intrusion detection on production systems: SIEM or cloud monitoring.

- Change management (CC8.1): Change policy and approvals.

  Usually from: GitHub settings, ticketing.

  - CC8.1: Change management policy, branch protection requiring reviews, separate test and production environments, deployment approvals and emergency change records: GitHub settings, CI/CD logs, ticketing.

  - Reviews required (SK-C3.5): your branch protection or ruleset settings from GitHub, showing required approving reviews, dismissal of stale approvals and who can bypass.

  - Direct pushes (SK-C3.6): the rule that blocks direct pushes to the default branch and who can bypass it, or the push audit log for the period.

- Risk mitigation and vendors (CC9.1, CC9.2): Business continuity plan, insurance, vendor inventory with risk ratings, and vendors' SOC 2 reports reviewed.

  Usually from: Vendor management in a compliance platform, contracts.

Contents

How to read this report . . . 2
What to do next . . . 2
Report details . . . 4
A. SOC 2 pre-assessment: about this part . . . 5
A0. Management's assertion . . . 5
A1. System description: items 1 to 9 (DC 1 to DC 9) . . . 5
A2. Control tests and results . . . 6
A3. Deviations and management responses . . . 9
A4. Readiness by criterion . . . 10
A5. Evidence lists an auditor samples from (populations) . . . 14
1. Executive summary . . . 15
2. Scope and scans included in report . . . 15
3. Policy evaluation . . . 15
4. Standards mapping: SOC 2 Trust Services Criteria 2017 TSC (revised points of focus 2022) . . . 16
5. Findings . . . 19
6. Exceptions and suppressions . . . 19
7. Trend and issues no longer detected . . . 19
8. Methodology and definitions . . . 20
9. Limitations and disclaimer . . . 21
10. Integrity and verification . . . 22

Report details

Prepared for	Example Co (fictional sample)
Generated at (UTC)	2026-11-10 09:00:00Z
Generated by	a workspace member
Report ID	00000000-0000-4000-8000-000000000002
Projects analysed	2 of 2 projects assessed (trusted, complete, current scan of the reported branch); 2 of 2 also ran every check the policy needs (security, secrets, dependencies); 2 of 2 have at least one scan.
Branch rule	Scans of each repository's default branch, as resolved by Skylos Cloud at upload.
Analyses included	Static security analysis (SAST); Secret detection; Dependency vulnerabilities (SCA); Quality analysis
Audit period	2026-08-13 to 2026-11-10 (90 days); issues no longer detected and scan history use it; findings and the policy result come from each project's latest assessed scan
Report SHA-256	c683c9d432c472cd2bbb350a513c75eb3f6e7799b0e61e1e3ff3d17bca3a7761
Signature	Unsigned: no signing key was configured in Skylos Cloud when this report was generated. The SHA-256 and the verification page still detect changes to the stored report.
Verify the stored record	The link at the foot of every page. Section 10 says what the verification page shows and how to check a copy.
Generator	Skylos Cloud build sample; mapping 2026-11-08.2; CLI rule catalog 5.6.0; renderer 2026-11-13.1

A. SOC 2 pre-assessment: about this part

Skylos pre-assessment, shaped like sections III to V of a SOC 2 Type 2 report. Skylos tested its own records; this is not an auditor's test, opinion or report.

Audit period: 2026-08-13 to 2026-11-10 (90 days), chosen when the report was generated. Period counts use only records from this period; findings and policy results are as of the generation time.

Trust services categories in scope: Security (33 criteria). A4 and section 4 list only their criteria.

Results read like a SOC 2 report's: "No deviations noted." or "Deviation noted: x of y ...". Where a disclosed Skylos limitation (named files or packages Skylos could not fully examine) left part of a population unexamined, the result adds "Unable to test: ...": a coverage exception of the tool, never a deviation of the company. The same wording is used where a scan record is no longer stored (removed by the plan's scan limit, a retention setting or a deletion): Skylos cannot tell what that scan showed, so the item is not tested and is not a deviation. Skylos ran these tests on its own records (scans, findings, gate results, GitHub pull-request records). An auditor selects samples from the populations and performs their own tests; this part gives no opinion.

A0. Management's assertion

Not provided by management. A SOC 2 report opens with management's written assertion, signed by an officer, that the system description is fairly presented and that the controls were suitably designed and operated effectively throughout the period. Skylos never writes or signs it.

Draft for management to adapt and sign (not provided by management): "We have prepared the accompanying description of Example Co (fictional sample)'s system for the period 2026-08-13 to 2026-11-10 (90 days), based on the description criteria in DC 200. We confirm, to the best of our knowledge and belief, that the description presents the system fairly; that the controls stated in it were suitably designed to provide reasonable assurance that our service commitments and system requirements would be achieved based on the applicable trust services criteria; and that those controls operated effectively throughout the period."

A1. System description: items 1 to 9 (DC 1 to DC 9)

Audit period: 2026-08-13 to 2026-11-10 (90 days)

Provided by management, not tested by Skylos. Management wrote 3 of 9 description items (DC 3: 1 of 5 parts). An item management has not written says "Not provided by management." Skylos writes none of them; the only Skylos text here is the DC 3 draft, labelled as such.

DC 1. Types of services provided

Fictional sample. Example Co provides a hosted order-tracking service to small businesses, with email support on business days.

DC 2. Principal service commitments and system requirements

Fictional sample. Customer agreements commit Example Co to protect customer data from unauthorized access and to fix CRITICAL and HIGH vulnerabilities within the deadlines of its security policy.

DC 3. Components of the system used to provide the services

Infrastructure: Not provided by management.

Software: Fictional sample. Two repositories (example-api, example-web): Python web applications with a PostgreSQL database, scanned by Skylos in CI.

People: Not provided by management.

Procedures: Not provided by management.

Data: Not provided by management.

Skylos draft for software and procedures, generated from Skylos records for the period and offered as a starting point (not written by management): The Company uses Skylos to perform static application security testing (SAST), secret detection and dependency vulnerability checks (SCA) of the source code in 2 repositories (example-api and example-web). From 2026-08-13 to 2026-11-10, 19 of 20 merged pull requests had a trusted Skylos scan before they were merged. The default branch of example-api was scanned by a trusted CI upload at least once every 7 days throughout that period (1 of 2 repositories in scope). For example-api, the Skylos check is required before a pull request can be merged into the default branch (GitHub setting read by Skylos on 2026-11-09). Findings rated CRITICAL must be fixed within 7 days and HIGH within 30 days of first detection. These are the Skylos default deadlines; the workspace has not set its own. An exception to a deadline is approved in Skylos Cloud with a reason, an approver and an expiry date; 1 active exception covered findings in scope when this report was generated.

Sources of the draft: Repositories and analyses: the checks each repository's latest trusted, complete default-branch scan ran. Pull requests: merges recorded by the Skylos GitHub App, matched to Skylos pull-request scans. Default branch: intervals between trusted Skylos scans of the default branch (scan_gaps.csv). Required check: default-branch protection Skylos read through its GitHub App. Exceptions: approved exception requests in Skylos Cloud.

Deadlines: the Skylos policy's fix deadlines (workspace settings or Skylos defaults).

DC 4. System incidents in the period

Not provided by management.

DC 5. Applicable trust services criteria and related controls

Not provided by management.

The controls Skylos tested are in A2 (with management's wording where management approved it), followed by the company controls management listed.

DC 6. Complementary user entity controls (CUECs)

Fictional sample. Customers manage their own users' access and protect their API keys.

DC 7. Subservice organizations and complementary subservice organization controls (carve-out method)

Not provided by management.

DC 8. Criteria not relevant to the system, with the reason

Not provided by management.

DC 9. Significant changes to the system in the period

Not provided by management.

A2. Control tests and results

Audit period: 2026-08-13 to 2026-11-10 (90 days)

Each control is stated as a company would state it, with the trust services criteria it supports. Skylos inspected its own records (inspection only; no inquiry, observation or reperformance) and tested every item in each population, not a sample. Tests marked "point in time" read one moment (a setting or the latest scan), not the whole period.

SK-C1. Code scanning on every pull request and on the default branch

Criteria: CC4.1; CC5.2; CC6.8; CC7.1; CC7.2. Control (management's wording, approved by management): Fictional sample. Skylos static analysis (security, secrets and dependency checks) runs on every pull request before merge and on the default branch at least every 7 days. Control owner (provided by management): Head of Engineering.

Test	Skylos test performed	Population	Result	Evidence file
SK-C1.1 Scan before merge	Inspected the Skylos scan records for the population of pull requests merged in the period, as recorded from GitHub by the Skylos GitHub App, to determine whether each pull request had a trusted Skylos scan (CI with OIDC or a CI key marked trusted) before it was merged.	20 merged pull requests in 2 repositories.	**Deviation noted: 1 of 20 merged pull requests had no trusted Skylos scan before merge.**	merged_prs.csv
SK-C1.2 Default-branch scan every week	Computed, for each in-scope repository, the time between consecutive trusted Skylos scans of its default branch, counting from the start of the period to the first scan and from the last scan to the end of the period, to determine whether the default branch was scanned at least once every 7 days (no interval longer than 7 days).	28 intervals in 2 repositories between 26 trusted default-branch scans in the period, from the start of the period to its end. Time before a repository was added to Skylos counts toward its first interval. Longest interval per repository: example-api 7.0 days; example-web 21.0 days.	**Deviation noted: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days).**	scan_gaps.csv

Test	Skylos test performed	Population	Result	Evidence file
SK-C1.3 Checks run in every scan	Inspected the check receipt Skylos stored for the last trusted scan before merge of each merged pull request and for every trusted scan of each in-scope repository's default branch in the period, to determine whether the scan ran the security, secrets and dependency checks and each one completed. Parts a check could not examine because of a disclosed Skylos limitation (named files or packages, within the disclosure limits) are stated as unable to test, not as deviations.	45 trusted scans: the last trusted scan before merge of 19 merged pull requests and 26 trusted default-branch scans in the period. A pull-request scan may cover only the changed files; that is not a deviation.	No deviations noted.	scan_checks.csv

SK-C2. Vulnerability remediation within fix-by dates, including dependency advisories

Criteria: CC7.1; CC7.2; CC4.2. Control (management's wording, approved by management): Fictional sample. CRITICAL findings are fixed within 7 days and HIGH within 30 days of first detection, or an exception is approved with a reason and an expiry date. Control owner (provided by management): Security lead.

Test	Skylos test performed	Population	Result	Evidence file
SK-C2.1 Fixes by the fix-by date	Inspected the population of CRITICAL and HIGH security, secret and dependency findings that were open at any time in the period in the repositories in scope, with their first-seen dates, fix-by dates, the first trusted, complete default-branch scan without the finding, and approved exceptions, to determine whether each was gone from a scan by its fix-by date or covered by an approved risk acceptance. This is the population the policy result and CC7.1 count. The first scan without the finding shows when Skylos stopped detecting it, not when a change was merged or deployed; an auditor samples commits, tickets or deployment records for that.	5 CRITICAL or HIGH findings in 2 repositories (1 dependency advisory): 1 with the first scan without the finding by the fix-by date, 0 after it, 3 open within the deadline (first due 2026-11-14), 0 open past it, 1 risk-accepted, 0 suppressed without an approval record.	No deviations noted.	findings.csv

SK-C3. Change management: pull requests scanned, gated and approved before merge

Criteria: CC8.1. Control (management's wording, approved by management): Fictional sample. Every change reaches the default branch through a pull request that Skylos scans and that someone other than the author approves. Control owner (provided by management): Head of Engineering.

Test	Skylos test performed	Population	Result	Evidence file
SK-C3.1 Gate result before merge	Inspected the last trusted Skylos scan before merge for the population of merged pull requests that had one, to determine whether the Skylos gate passed, or failed and was overridden before the merge.	19 merged pull requests with a trusted Skylos scan before merge: 18 passed, 1 failed then overridden, 0 failed, 0 gate off. Merged pull requests only: changes pushed straight to the default branch are not in this population (see SK-C3.6).	No deviations noted.	merged_prs.csv
SK-C3.2 Override records	Inspected the override record for the population of merged pull requests whose last Skylos gate before merge failed and was overridden, to determine whether each override recorded who overrode it, when and why.	1 overridden gate before merge. Only workspace owners and admins can override. Merged pull requests only: changes pushed straight to the default branch are not in this population (see SK-C3.6).	No deviations noted.	merged_prs.csv

Test	Skylos test performed	Population	Result	Evidence file
SK-C3.3 Approval by another person	Inspected the GitHub reviews Skylos recorded for the population of merged pull requests to determine whether each was approved before merge by a person other than its author (bots and AI reviewers not counted).	20 merged pull requests with recorded GitHub reviews. Merged pull requests only: changes pushed straight to the default branch are not in this population (see SK-C3.6).	**Deviation noted: 1 of 20 merged pull requests had no approval from a person other than the author before merge.**	merged_prs.csv
SK-C3.4 Skylos check required	Inspected the default-branch protection Skylos last read from GitHub for each in-scope repository connected through the Skylos GitHub App to determine whether the Skylos check is a required status check. Point in time: the setting when read, not a history over the period.	2 repositories. Point in time: read from GitHub on 2026-11-09; this is the setting when read, not throughout the period.	**Deviation noted: the Skylos check is not required on the default branch of 1 of 2 repositories (example-web).**	branch_settings.csv
SK-C3.5 Reviews required	Branch protection requiring an approving review before merge.	Not available	Not evidenced by Skylos: Skylos reads whether its own check is required, not whether reviews are required. Provide the default-branch protection or ruleset settings from GitHub (required approving reviews, dismissal of stale approvals, who can bypass).	None: not evidenced by Skylos
SK-C3.6 Direct pushes	Changes that reached the default branch without a pull request (direct pushes).	Not available	Not evidenced by Skylos: SK-C3.1 to SK-C3.3 cover merged pull requests only. Skylos does not record which changes reached the default branch without a pull request: its merged pull request records carry no merge commit, so a default-branch scan cannot be matched to one. Provide the default-branch protection rule or ruleset that blocks direct pushes (and who can bypass it), or the repository's push audit log for the period.	None: not evidenced by Skylos

SK-C4. CI/CD pipeline hardening

Criteria: CC6.8; CC8.1. Control (Skylos draft wording, not approved by management): CI/CD pipeline definitions on the default branch have no open CRITICAL or HIGH weaknesses (for example unpinned third-party actions, broad token permissions, untrusted input in run steps).

Test	Skylos test performed	Population	Result	Evidence file
SK-C4.1 CI/CD pipeline findings	Inspected the CI/CD pipeline findings of the latest assessed scan of each in-scope repository (point in time: that scan, at generation) to determine whether pipeline definitions on the default branch had open CRITICAL or HIGH findings.	2 assessed repositories that ran the security check. Point in time: the latest assessed scan of each, at generation. Open CI/CD pipeline findings: 0 CRITICAL, 0 HIGH, 0 MEDIUM, 0 LOW. Only CRITICAL and HIGH count as deviations. Skylos does not record whether a repository has pipeline files; one without them has no findings here.	No deviations noted.	cicd_findings.csv

SK-C5. No hard-coded secrets on the default branch

Criteria: CC6.1. Control (Skylos draft wording, not approved by management): Credentials, keys and tokens are not stored in source code on the default branch.

Test	Skylos test performed	Population	Result	Evidence file
SK-C5.1 Hard-coded secrets	Inspected the secret findings of the latest assessed scan of each in-scope repository (point in time: that scan, at generation) to determine whether the default branch had open CRITICAL or HIGH hard-coded secrets.	2 assessed repositories that ran the secrets check. Point in time: the latest assessed scan of each, at generation. Open secret findings: 0 CRITICAL, 0 HIGH, 0 MEDIUM, 0 LOW; 1 suppressed with an approved exception. Only CRITICAL and HIGH count as deviations. Secrets in git history before the scanned commit are not covered.	No deviations noted.	secret_findings.csv

SK-C6. Dependency inventory

Criteria: CC6.8; CC7.1. Control (Skylos draft wording, not approved by management): Third-party dependencies of each repository are inventoried and checked against public vulnerability advisories by the Skylos dependency check.

Test	Skylos test performed	Population	Result	Evidence file
SK-C6.1 Dependency check coverage	Inspected the dependency coverage receipt of the latest assessed scan of each in-scope repository (point in time: that scan, at generation; SK-C1.3 covers every scan in the period) to determine whether its declared dependencies were inventoried and checked against public advisories (OSV) and the check completed. Dependencies Skylos names as ones it could not fully examine (a disclosed Skylos limitation) are stated as unable to test, not as deviations.	2 assessed repositories (example-api: 42 queried; example-web: 42 queried). Point in time: the latest assessed scan of each, at generation.	No deviations noted.	dependency_coverage.csv
SK-C6.2 SBOM per release	A software bill of materials (SBOM) for each release.	Not available	Not evidenced by Skylos: Skylos Cloud does not store an SBOM. The Skylos CLI writes a CycloneDX SBOM (skylos sbom); provide it, or one from your build system, for the releases in the period.	None: not evidenced by Skylos

Company controls (provided by management, not tested by Skylos)

Management listed no company controls. Controls Skylos does not test (for example security training, access reviews, vendor reviews) can be added in Skylos Cloud.

A3. Deviations and management responses

Audit period: 2026-08-13 to 2026-11-10 (90 days)

Other information provided by management, not tested by Skylos: a response is management's corrective action recorded in Skylos (for one deviation, or for every deviation of a test) and any recorded risk acceptance. Skylos never writes a response; where none was recorded the row says so. The summary counts every deviation; the tables list the first items of each test, and deviations.csv lists all of them with their responses.

Test	Criteria	Result	Items	With a management response
SK-C1.1	CC4.1; CC5.2; CC6.8; CC7.1; CC7.2	Deviation noted: 1 of 20 merged pull requests had no trusted Skylos scan before merge.	1	0
SK-C1.2	CC4.1; CC5.2; CC6.8; CC7.1; CC7.2	Deviation noted: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days).	1	0
SK-C3.3	CC8.1	Deviation noted: 1 of 20 merged pull requests had no approval from a person other than the author before merge.	1	0

Test	Criteria	Result	Items	With a management response
SK-C3.4	CC8.1	Deviation noted: the Skylos check is not required on the default branch of 1 of 2 repositories (example-web).	1	1

Management's responses for every deviation of a test

Test	Corrective action (provided by management)	Owner	Due date
SK-C3.4	Corrective action (for every C3.4 deviation), provided by management: Fictional sample. The Skylos check is being added to the required status checks of example-web's default branch. Owner: Head of Engineering. Due 2026-11-30.	Head of Engineering	2026-11-30

SK-C1.1 (CC4.1; CC5.2; CC6.8; CC7.1; CC7.2)

Project, item	Deviation noted	Management response
example-web example/example-web #105 (merged 2026-09-30)	No Skylos scan before merge	Management response: not provided

SK-C1.2 (CC4.1; CC5.2; CC6.8; CC7.1; CC7.2)

Project, item	Deviation noted	Management response
example-web 21.0 days without a trusted scan: 2026-09-12 (scan) to 2026-10-03 (scan)	No trusted scan of the default branch for more than 7 days	Management response: not provided

SK-C3.3 (CC8.1)

Project, item	Deviation noted	Management response
example-api example/example-api #109 (merged 2026-10-12)	No approval from a person other than the author (example-dev-a) before merge	Management response: not provided

SK-C3.4 (CC8.1)

Project, item	Deviation noted	Management response
example-web example/example-web	Skylos check not required on the default branch (read 2026-11-09)	Management's corrective action for every SK-C3.4 deviation (above).

A4. Readiness by criterion

Audit period: 2026-08-13 to 2026-11-10 (90 days)

This table lists 9 criteria: 8 that a Skylos control test (A2) is cross-listed to, and 1 with findings mapped in section 4 but no control test (CC6.7). The status has one basis, the A2 tests; an open CRITICAL or HIGH finding mapped to the criterion also keeps it at "Gaps found". Other mapped findings are a note. A passed test covers only the part of a criterion named in A2, never the whole criterion. Section 1 counts criteria with mapped findings or scan activity (4 of 33).

Skylos test results (8 criteria): passed for 2, gaps found for 6, failed for 0, not evidenced for 0. Findings only, no status: 1. The rest of every criterion is not evidenced by Skylos (last columns); 24 more criteria in scope need evidence Skylos does not have (grouped below).

Management tracker (provided by management, not tested by Skylos), 33 criteria in scope: management status for 2; priority for 2; owner for 2; target date for 2; written implementation response for 2. The Skylos test result above is not changed by it. Needs reconciliation: none (no criterion management marks Implemented has a Skylos deviation or an open CRITICAL or HIGH finding capping it). Every criterion is in readiness.csv.

A4 index: one row per criterion

Every criterion in scope on one page. Skylos tests: the result of the Skylos tests cross-listed to the criterion (A2), "Findings only" when only section 4 findings map to it, "Not tested by Skylos" when Skylos has no evidence for it. The detail follows.

Criterion	Skylos tests	Management status	Reconciliation	Owner	Target date
CC1.1	Not tested by Skylos	Not provided			
CC1.2	Not tested by Skylos	Not provided			
CC1.3	Not tested by Skylos	Not provided			
CC1.4	Not tested by Skylos	Not provided			
CC1.5	Not tested by Skylos	Not provided			
CC2.1	Not tested by Skylos	Not provided			
CC2.2	Not tested by Skylos	Not provided			
CC2.3	Not tested by Skylos	Not provided			
CC3.1	Not tested by Skylos	Not provided			
CC3.2	Not tested by Skylos	Not provided			
CC3.3	Not tested by Skylos	Not provided			
CC3.4	Not tested by Skylos	Not provided			
CC4.1	**Gaps found**	Not provided			
CC4.2	**Tests passed**	Not provided			
CC5.1	Not tested by Skylos	Not provided			
CC5.2	**Gaps found**	Not provided			
CC5.3	Not tested by Skylos	Not provided			
CC6.1	**Tests passed**	In progress		Security lead	2026-12-15
CC6.2	Not tested by Skylos	Not provided			
CC6.3	Not tested by Skylos	Not provided			
CC6.4	Not tested by Skylos	Not provided			
CC6.5	Not tested by Skylos	Not provided			
CC6.6	Not tested by Skylos	Not provided			
CC6.7	**Findings only**	Not provided			
CC6.8	**Gaps found**	Not provided			
CC7.1	**Gaps found**	Not provided			
CC7.2	**Gaps found**	Not provided			
CC7.3	Not tested by Skylos	Not provided			
CC7.4	Not tested by Skylos	Not provided			
CC7.5	Not tested by Skylos	Not provided			
CC8.1	**Gaps found**	In progress		Head of Engineering	2026-11-30
CC9.1	Not tested by Skylos	Not provided			
CC9.2	Not tested by Skylos	Not provided			

A4 detail

Criterion	Skylos test result	Management tracker (not tested by Skylos)	Gap in the tested controls	Recommended action	Rest of the criterion
CC4.1 Ongoing and separate evaluations confirm controls are present and functioning Tests: SK-C1.1, SK-C1.2, SK-C1.3	**Gaps found: SK-C1.1, SK-C1.2, SK-C1.3**	Not provided by management.	SK-C1.1: 1 of 20 merged pull requests had no trusted Skylos scan before merge. SK-C1.2: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days).	Run Skylos on every pull request (the Skylos GitHub workflow) and make the Skylos check required on the default branch. Scan the default branch on every push and on a weekly schedule.	Not evidenced by Skylos. Needs: How the other controls are monitored: compliance-platform monitoring, internal audit or control self-assessments.

Criterion	Skylos test result	Management tracker (not tested by Skylos)	Gap in the tested controls	Recommended action	Rest of the criterion
CC4.2 Internal control deficiencies are evaluated and communicated Tests: SK-C2.1	**Skylos tests passed: SK-C2.1**	Not provided by management.	No gap in the controls Skylos tests.	Keep the controls operating through the audit period.	Not evidenced by Skylos. Needs: How control deficiencies reach management and are tracked to closure: tickets, meeting minutes.
CC5.2 General control activities over technology are selected and developed Tests: SK-C1.1, SK-C1.2, SK-C1.3	**Gaps found: SK-C1.1, SK-C1.2, SK-C1.3**	Not provided by management.	SK-C1.1: 1 of 20 merged pull requests had no trusted Skylos scan before merge. SK-C1.2: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days).	Run Skylos on every pull request (the Skylos GitHub workflow) and make the Skylos check required on the default branch. Scan the default branch on every push and on a weekly schedule.	Not evidenced by Skylos. Needs: Technology controls beyond code: access, infrastructure and operations policies and their operation.
CC6.1 Logical access security software, infrastructure and architectures protect information assets Tests: SK-C5.1	**Skylos tests passed: SK-C5.1**	In progress, priority High, owner Security lead, target 2026-12-15. Response: Fictional sample. The remaining hard-coded secrets are being moved to the secret manager and rotated.	No gap in the controls Skylos tests. Note (not a gap in the tested controls): also 1 open finding mapped to CC6.1 in section 4 (not a control test).	Fix or risk-accept the open findings mapped to CC6.1.	Not evidenced by Skylos. Needs: MFA and unique IDs on critical systems, access provisioning and periodic access reviews, encryption key management: identity provider and cloud console settings.
CC6.7 Transmission, movement and removal of information is restricted and protected Tests: none (mapped findings only)	**Findings only (no Skylos control test)**	Not provided by management.	No Skylos control test covers CC6.7, so it has no status here; 0 open findings mapped to CC6.7 in section 4, which does not show it is met.	Provide this evidence from the sources listed.	Not evidenced by Skylos. Needs: Encryption in transit and at rest, endpoint and removable-media encryption: cloud console and device-management settings.
CC6.8 Controls prevent or detect unauthorized or malicious software Tests: SK-C1.1, SK-C1.2, SK-C1.3, SK-C4.1, SK-C6.1, SK-C6.2	**Gaps found: SK-C1.1, SK-C1.2, SK-C1.3, SK-C4.1, SK-C6.1, SK-C6.2**	Not provided by management.	SK-C1.1: 1 of 20 merged pull requests had no trusted Skylos scan before merge. SK-C1.2: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days). SK-C6.2: not evidenced by Skylos: Skylos Cloud does not store an SBOM. The Skylos CLI writes a CycloneDX SBOM (skylos sbom); provide it, or one from your build system, for the releases in the period.	Run Skylos on every pull request (the Skylos GitHub workflow) and make the Skylos check required on the default branch. Scan the default branch on every push and on a weekly schedule.	Not evidenced by Skylos. Needs: Endpoint anti-malware, operating-system patching and approved software: device management (MDM) and endpoint protection reports.
CC7.1 Detection and monitoring procedures identify configuration changes that introduce vulnerabilities and susceptibility to newly discovered vulnerabilities Tests: SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1, SK-C6.1, SK-C6.2	**Gaps found: SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1, SK-C6.1, SK-C6.2**	Not provided by management.	SK-C1.1: 1 of 20 merged pull requests had no trusted Skylos scan before merge. SK-C1.2: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days). SK-C6.2: not evidenced by Skylos: Skylos Cloud does not store an SBOM. The Skylos CLI writes a CycloneDX SBOM (skylos sbom); provide it, or one from your build system, for the releases in the period.	Run Skylos on every pull request (the Skylos GitHub workflow) and make the Skylos check required on the default branch. Scan the default branch on every push and on a weekly schedule.	Not evidenced by Skylos. Needs: Infrastructure vulnerability scans, penetration test reports and production configuration monitoring.

Criterion	Skylos test result	Management tracker (not tested by Skylos)	Gap in the tested controls	Recommended action	Rest of the criterion
CC7.2 System components are monitored for anomalies that indicate malicious acts, disasters or errors Tests: SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1	**Gaps found: SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1**	Not provided by management.	SK-C1.1: 1 of 20 merged pull requests had no trusted Skylos scan before merge. SK-C1.2: 1 of 28 intervals between trusted scans of the default branch was longer than 7 days, in 1 of 2 repositories (example-web: longest 21.0 days, 1 over 7 days).	Run Skylos on every pull request (the Skylos GitHub workflow) and make the Skylos check required on the default branch. Scan the default branch on every push and on a weekly schedule.	Not evidenced by Skylos. Needs: Logging, alerting and intrusion detection on production systems: SIEM or cloud monitoring.
CC8.1 Changes to infrastructure, data, software and procedures are authorized, designed, developed, tested, approved and implemented Tests: SK-C3.1, SK-C3.2, SK-C3.3, SK-C3.4, SK-C3.5, SK-C3.6, SK-C4.1	**Gaps found: SK-C3.1, SK-C3.2, SK-C3.3, SK-C3.4, SK-C3.5, SK-C3.6, SK-C4.1**	In progress, priority Medium, owner Head of Engineering, target 2026-11-30. Response: Fictional sample. The Skylos check is being made a required status check on example-web.	SK-C3.3: 1 of 20 merged pull requests had no approval from a person other than the author before merge. SK-C3.4: the Skylos check is not required on the default branch of 1 of 2 repositories (example-web). SK-C3.5: not evidenced by Skylos: Skylos reads whether its own check is required, not whether reviews are required. Provide the default-branch protection or ruleset settings from GitHub (required approving reviews, dismissal of stale approvals, who can bypass). SK-C3.6: not evidenced by Skylos: SK-C3.1 to SK-C3.3 cover merged pull requests only. Skylos does not record which changes reached the default branch without a pull request: its merged pull request records carry no merge commit, so a default-branch scan cannot be matched to one. Provide the default-branch protection rule or ruleset that blocks direct pushes (and who can bypass it), or the repository's push audit log for the period.	Require an approving review from someone other than the author in branch protection. Add the Skylos check to the required status checks of the default branch.	Not evidenced by Skylos. Needs: Change management policy, branch protection requiring reviews, separate test and production environments, deployment approvals and emergency change records: GitHub settings, CI/CD logs, ticketing.

Criteria Skylos does not evidence

Area	Criteria	What the auditor will ask for	Where it usually comes from
Control environment	CC1.1, CC1.2, CC1.3, CC1.4, CC1.5	Code of conduct and its acknowledgements, organization chart, management or board oversight, background checks, security training and performance reviews.	HR system, policy documents, a compliance platform (for example Vanta, Drata or Secureframe).
Communication and information	CC2.1, CC2.2, CC2.3	Security policies and how they are communicated internally; the system description and commitments communicated to customers; a channel to report security issues.	Policy documents, the internal wiki, customer contracts and the security page.
Risk assessment	CC3.1, CC3.2, CC3.3, CC3.4	An annual risk assessment, a risk register with owners and treatments, fraud risk, and how significant changes are assessed.	Risk register (spreadsheet or GRC tool), management meeting minutes.
Monitoring and control activities	CC5.1, CC5.3	Control activities chosen for the risks, and the policies and procedures that put them in place.	Policy documents, a compliance platform.
Logical and physical access	CC6.2, CC6.3, CC6.4, CC6.5, CC6.6	User provisioning and removal records, periodic access reviews, MFA and SSO settings, firewall and WAF rules, physical security, asset disposal.	Identity provider (for example Okta or Google Workspace), cloud console, HR offboarding records; physical security usually from the cloud provider's SOC 2 report (carve-out).
System operations and incidents	CC7.3, CC7.4, CC7.5	Security event triage, incident classification, response and recovery records.	Incident tickets, the on-call tool, post-incident reviews.
Risk mitigation and vendors	CC9.1, CC9.2	Business continuity plan, insurance, vendor inventory with risk ratings, and vendors' SOC 2 reports reviewed.	Vendor management in a compliance platform, contracts.

Management tracker for the other criteria

Management tracks none of the other criteria yet.

A5. Evidence lists an auditor samples from (populations)

Audit period: 2026-08-13 to 2026-11-10 (90 days)

Lists an auditor selects samples from, as CSV downloads that carry this report's id and SHA-256.

File	Rows	Evidence for	What it lists
merged_prs.csv	20	SK-C1.1, SK-C3.1, SK-C3.2, SK-C3.3	Pull requests merged in the period, with the Skylos scans before merge, the gate result, any override and the approvers other than the author.
scan_checks.csv	45	SK-C1.3	The last trusted scan before merge of each merged pull request and every trusted default-branch scan in the period, with how each ran the security, secrets and dependency checks.
weekly_scans.csv	26	SK-C1.2 (by week)	Trusted default-branch scans and pull-request scans per repository per week of the period.
scan_gaps.csv	28	SK-C1.2	Every interval between trusted default-branch scans per repository, from the start of the period to its end, with its length and whether it is longer than 7 days.
findings.csv	5	SK-C2.1	CRITICAL and HIGH security, secret and dependency findings open at any time in the period, with first seen and fix-by dates, the first scan without the finding (when Skylos stopped detecting it, not a merge or deploy date), status at the end of the period and any risk acceptance.
cicd_findings.csv	0	SK-C4.1	Open and suppressed CI/CD pipeline findings of the latest assessed scans, every severity (point in time).
secret_findings.csv	1	SK-C5.1	Open and suppressed secret findings of the latest assessed scans, every severity (point in time).
branch_settings.csv	2	SK-C3.4	Whether the Skylos check is a required status check on each repository's default branch, as last read from GitHub (point in time).
dependency_coverage.csv	2	SK-C6.1	The dependency coverage receipt of each assessed repository's latest assessed scan (point in time).
deviations.csv	4	A3	Every deviation of the tests in A2, with the management response recorded in Skylos, then every coverage exception (unable to test), each with its kind.
readiness.csv	33	A4	Every criterion in scope: the result of the Skylos tests cross-listed to it (or the area Skylos does not evidence) next to management's tracker (status, priority, owner, target date, implementation response), which is provided by management and not tested by Skylos.

- branch_settings.csv: Point in time: each setting as Skylos last read it from GitHub, not throughout the period.
- dependency_coverage.csv: Point in time: the latest assessed scan of each repository, at generation. scan_checks.csv covers every scan in the period.

Compliance platforms such as Drata and Vanta accept the PDF and these CSV files as custom evidence uploads. You upload them yourself; Skylos does not connect to those platforms.

Download links (sign in to Skylos Cloud first), one per file in the order above:

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=merged_prs&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=scan_checks&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=weekly_scans&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=scan_gaps&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=findings&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=cicd_findings&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=secret_findings&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=branch_settings&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=dependency_coverage&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=deviations&edition=external&download=1

https://skylos.dev/api/compliance/reports/00000000-0000-4000-8000-000000000002?format=csv&population=readiness&edition=external&download=1

1. Executive summary

Policy status (Skylos default security policy, revision 2):

Policy met, fixes due

- 3 open CRITICAL or HIGH findings are within the fix deadline; none is overdue.

Before exceptions the policy status would be Policy not met. The status is about the stated policy, not a determination about SOC 2 Trust Services Criteria.

Open in this report	CRITICAL	HIGH	Past fix-by	Suppressed	Active exceptions	New since previous	No longer detected in the period
4	1	2	0	1 (20%)	1	-	1

Requirements listed: 33. Open findings: 1; No open findings from mapped rules: 2; Scan activity recorded: 1; Deviation noted: 0; Not scanned: 0; Not evidenced by static analysis: 29.

Skylos scans give evidence toward 4 of the 33 listed requirements of SOC 2 Trust Services Criteria 2017 TSC (revised points of focus 2022) (through mapped findings or scan activity); that is not proof they are met. The other 29 need manual, runtime or process evidence, or no Skylos rule checks them, and are listed with the reason. The 33 common criteria (Security), the category selected for this report. Part A's readiness table (A4) lists 9 criteria: 8 that its control tests (A2) are cross-listed to, with a status for the tested controls, and 1 with mapped findings only (CC6.7), without a status.

2. Scope and scans included in report

2 of 2 projects assessed (trusted, complete, current scan of the reported branch); 2 of 2 also ran every check the policy needs (security, secrets, dependencies); 2 of 2 have at least one scan.

Project	Business criticality	Assessment	Scan, branch, commit	CLI, completed, how uploaded	Coverage
example-api example/example-api	High	**Assessed**	c47617ea-ee98-4cce-83e8-a5664a4cd767 main 761ccd2a5ff0bba9875cdeebd7c0f0370e8c6c35	5.6.0 2026-11-10 06:30:00Z Verified CI (GitHub OIDC)	complete (full repository); checks: security, secrets, dependencies, quality, dead_code; dependencies: 42 queried
example-api: Trusted (CI with GitHub OIDC), complete scan of the default branch.					
example-web example/example-web	Medium	**Assessed**	ebb16937-3208-48d8-845a-8f4bdb2cab30 main 7869ab6ed164fe7245bd2c2ace0cd1fa5c5c0ff2	5.6.0 2026-11-10 06:30:00Z Verified CI (GitHub OIDC)	complete (full repository); checks: security, secrets, dependencies, quality, dead_code; dependencies: 42 queried
example-web: Trusted (CI with GitHub OIDC), complete scan of the default branch.					

Not included in report

- AI-generated code defect checks: not run in the selected scans
- Deep scan (VulnHunter, optional): not run in the selected scans
- Dynamic testing (DAST): not performed by Skylos
- Manual penetration test: not performed by Skylos
- Third-party tool results (uploaded SARIF): not included
- Results of third-party tools uploaded as SARIF are not included.
- A software component inventory (SBOM) and license findings are not included.
- Reachability and exploitability of dependency vulnerabilities are not assessed in this report.

3. Policy evaluation

Policy	Skylos default security policy, revision 2
Policy status	Policy met, fixes due

Statement	No open CRITICAL or HIGH security, secret or dependency finding past its fix-by date, with every in-scope project assessed by a trusted, complete scan of the reported branch, no older than 90 days, that ran security, secrets and dependency checks. Small gaps the CLI names in full (at most 50 files and 2% of the files analysed, or of the dependencies checked) are disclosed limitations, not a reason to refuse the scan.
Fix deadlines	Skylos default fix deadlines (CRITICAL 7 days, HIGH 30 days from first seen); the workspace has not turned on fix deadlines.

Rules

Rule	Requirement	Findings	Status
Maximum severity past deadline	No open CRITICAL or HIGH security, secret or dependency finding past its fix-by date	0 findings past the fix-by date	**Met**
Assessment coverage	Every in-scope project has a trusted, complete scan of the reported branch, no older than 90 days, that ran security, secrets, dependencies checks	2 of 2 projects meet it	**Met**

Scan requirements

Project	Frequency	Last performed	Assessment-freshness deadline	Status	Why
example-api	At least every 90 days	2026-11-10	2027-02-08	**Met**	
example-web	At least every 90 days	2026-11-10	2027-02-08	**Met**	

Remediation

Severity	Fix deadline	Findings exceeding	Status
CRITICAL	7 days (Skylos default)	0	**Met**
HIGH	30 days (Skylos default)	0	**Met**
MEDIUM	None set	0	**Not covered by policy**
LOW	None set	0	**Not covered by policy**

4. Standards mapping: SOC 2 Trust Services Criteria 2017 TSC (revised points of focus 2022)

The 33 common criteria (Security), the category selected for this report. Mappings are Skylos's own (mapping 2026-11-08.2).

Requirement	Title	Status	Open	Suppr.	Reason
CC1.1	Commitment to integrity and ethical values	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC1.2	Board independence and oversight of internal control	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC1.3	Structures, reporting lines, authorities and responsibilities	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC1.4	Commitment to attract, develop and retain competent individuals	**Not evidenced by static analysis**	0	0	People control: it needs HR, contract or training records.
CC1.5	Individuals held accountable for internal control responsibilities	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC2.1	Relevant, quality information is obtained or generated to support internal control	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC2.2	Internal communication of internal control information	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.

Requirement	Title	Status	Open	Suppr.	Reason
CC2.3	Communication with external parties about internal control	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC3.1	Objectives are specified clearly enough to identify and assess risks	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC3.2	Risks to objectives are identified and analyzed	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC3.3	Potential for fraud is considered in assessing risks	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC3.4	Changes that could significantly affect internal control are identified and assessed	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC4.1	Ongoing and separate evaluations confirm controls are present and functioning	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code. Part A tests the Skylos controls for part of CC4.1 (SK-C1.1, SK-C1.2, SK-C1.3): Gaps found in A4.
CC4.2	Internal control deficiencies are evaluated and communicated	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code. Part A tests the Skylos controls for part of CC4.2 (SK-C2.1): Skylos tests passed in A4.
CC5.1	Control activities are selected and developed to mitigate risks	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC5.2	General control activities over technology are selected and developed	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code. Part A tests the Skylos controls for part of CC5.2 (SK-C1.1, SK-C1.2, SK-C1.3): Gaps found in A4.
CC5.3	Control activities are deployed through policies and procedures	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC6.1	Logical access security software, infrastructure and architectures protect information assets	**Open findings**	1	1	1 open finding in 1 project. Mapped findings are code weaknesses that bypass or weaken access control, store or expose credentials or keys, or protect data with weak or broken cryptography. Part A tests the Skylos controls for part of CC6.1 (SK-C5.1): Skylos tests passed in A4.
CC6.2	New users are registered and authorized before credentials are issued	**Not evidenced by static analysis**	0	0	Operational control: it needs evidence from running systems, networks, endpoints or providers, not source code.
CC6.3	Access is authorized, modified and removed based on roles and least privilege	**Not evidenced by static analysis**	0	0	Operational control: it needs evidence from running systems, networks, endpoints or providers, not source code.
CC6.4	Physical access to facilities and protected assets is restricted	**Not evidenced by static analysis**	0	0	Physical control: it needs facility, equipment or media evidence.
CC6.5	Logical and physical protections are discontinued when assets are disposed of	**Not evidenced by static analysis**	0	0	Physical control: it needs facility, equipment or media evidence.
CC6.6	Logical access security measures protect against threats from outside system boundaries	**Not evidenced by static analysis**	0	0	Operational control: it needs evidence from running systems, networks, endpoints or providers, not source code.
CC6.7	Transmission, movement and removal of information is restricted and protected	**No open findings from mapped rules**	0	0	No open findings from the 6 mapped rules that ran in all 2 in-scope projects. Mapped findings disable certificate validation or send data or session cookies without transport protection.

Requirement	Title	Status	Open	Suppr.	Reason
CC6.8	Controls prevent or detect unauthorized or malicious software	**No open findings from mapped rules**	0	0	No open findings from the 13 mapped rules that ran in all 2 in-scope projects. 2 more mapped rules are in checks that did not run in every project (ai_defects). Mapped findings are code that downloads or runs unverified third-party code, or depends on untrustworthy components. Part A tests the Skylos controls for part of CC6.8 (SK-C1.1, SK-C1.2, SK-C1.3, SK-C4.1, SK-C6.1, SK-C6.2): Gaps found in A4.
CC7.1	Detection and monitoring procedures identify configuration changes that introduce vulnerabilities and susceptibility to newly discovered vulnerabilities	**Scan activity recorded**	4	1	No deviations noted. Population: 5 CRITICAL or HIGH findings in 2 repositories (1 dependency advisory): 1 with the first scan without the finding by the fix-by date, 0 after it, 3 open within the deadline (first due 2026-11-14), 0 open past it, 1 risk-accepted, 0 suppressed without an approval record. All 2 in-scope projects have a trusted, current scan that ran security, secrets, dependencies checks. Skylos evidences the code part of vulnerability management: trusted scans that find vulnerabilities in first-party code, secrets and third-party dependencies, and whether CRITICAL and HIGH ones are gone from a trusted scan by their fix-by dates (the first scan without the finding: when Skylos stopped detecting it, not a merge or deploy date). The test covers the policy's findings: every CRITICAL and HIGH security, secret and dependency finding, with or without a CWE id. Finding a vulnerability is this control operating; a deviation is one left open past its fix-by date, with its first scan without the finding after it, or suppressed without an approval record. It does not evidence infrastructure or host vulnerability scans, monitoring of running systems for configuration changes, penetration tests, or management's review of results. Part A tests the Skylos controls for part of CC7.1 (SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1, SK-C6.1, SK-C6.2): Gaps found in A4.
CC7.2	System components are monitored for anomalies that indicate malicious acts, disasters or errors	**Not evidenced by static analysis**	0	0	Operational control: it needs evidence from running systems, networks, endpoints or providers, not source code. Part A tests the Skylos controls for part of CC7.2 (SK-C1.1, SK-C1.2, SK-C1.3, SK-C2.1): Gaps found in A4.
CC7.3	Security events are evaluated to determine whether they are incidents	**Not evidenced by static analysis**	0	0	Operational control: it needs evidence from running systems, networks, endpoints or providers, not source code.
CC7.4	Security incidents are responded to with a defined incident-response program	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC7.5	Recovery from identified security incidents	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.

Requirement	Title	Status	Open	Suppr.	Reason
CC8.1	Changes to infrastructure, data, software and procedures are authorized, designed, developed, tested, approved and implemented	**Not evidenced by static analysis**	0	0	Static analysis findings do not evidence change management. Per-change evidence for merged pull requests (the Skylos scan before merge, the gate result, any override and its reason, approval by someone other than the author) is in Part A of this report: tests SK-C3.1 to SK-C3.3 (A2) and merged_prs.csv (A5). Changes pushed straight to the default branch are not evidenced by Skylos (SK-C3.6). Part A tests the Skylos controls for part of CC8.1 (SK-C3.1, SK-C3.2, SK-C3.3, SK-C3.4, SK-C3.5, SK-C3.6, SK-C4.1): Gaps found in A4.
CC9.1	Risk mitigation activities for potential business disruptions	**Not evidenced by static analysis**	0	0	Organizational control: it needs policies, roles, management decisions or records, not source code.
CC9.2	Vendor and business partner risk is assessed and managed	**Not evidenced by static analysis**	0	0	Supplier or contract control: it needs agreements and supplier reviews.

5. Findings

The external summary lists counts only; finding locations and text are in the full edition.

Other findings not mapped to this framework: none.

Severity	Open	Past fix-by date
CRITICAL	1	0
HIGH	2	0
MEDIUM	1	0
LOW	0	0
UNKNOWN	0	0

6. Exceptions and suppressions

1 of 5 findings in this report suppressed (20%), 0 without an approval record. 1 active exception, 1 expiring within 30 days, 0 without an end date. Verdict before exceptions: Policy not met; after: Policy met, fixes due.

7. Trend and issues no longer detected

No earlier report of this framework and scope exists. Issues no longer detected below cover the audit period.

Scan history in the window

Trusted scans of each project's report branch from 2026-08-13 to 2026-11-10, with the longest stretch without one. Scans removed by your retention setting are not counted.

Project	Trusted scans	First	Last	Longest stretch without one
example-api	14	2026-08-15 00:00:00Z	2026-11-10 06:30:00Z	7 days
example-web	12	2026-08-15 00:00:00Z	2026-11-10 06:30:00Z	21 days

No longer detected in the window

1 of 1 CRITICAL or HIGH issue no longer detected in the window had the first scan without the finding by the fix-by date; 0 after it. That date shows when Skylos stopped detecting the finding, not when a change was merged or deployed.

Project, rule	Severity	First scan without the finding	By the fix-by date	Scan, commit
example-api SKY-D212	HIGH	2026-10-29 09:00:00Z	Yes (due 2026-11-10)	4ec5299c-ef87-4e43-8fc7-edb06a35bcba 13ffc9a7f1deedd1015ceafa8322ec929a9d5f22

8. Methodology and definitions

Analysis	What it does	What it cannot see
Static security analysis (SAST)	Skylos security rules on first-party code, configuration, CI pipelines and infrastructure files.	Runtime behaviour, deployed configuration and business logic.
Secret detection	Hard-coded credentials, keys and tokens in the scanned files.	Secrets in git history before the scanned commit, or outside the repository.
Dependency vulnerabilities (SCA)	Declared dependencies matched against public advisories (OSV) when the scan ran.	Whether a vulnerable function is reachable, and dependencies without a supported manifest or resolved version.
Quality analysis	Reliability and maintainability rules; counted here only where a rule maps to a requirement.	Security weaknesses outside its rule set.
AI-generated code defect checks	Hallucinated packages and APIs, weakened tests and other AI-agent mistakes.	Defects without a Skylos rule.
Deep scan (VulnHunter, optional)	An AI security review of the repository, each finding challenged by a second AI pass.	Anything outside its review budget; partial scans are labelled.

Staleness rule. Results from a scan older than 90 days at generation time do not count toward "Policy met": the project is shown as not assessed and its next scan is overdue.

Mapping. Findings map to requirements through CWE ids declared by the Skylos CLI (catalog 5.6.0), CWE ids Skylos Cloud assigns to 63 rules the CLI ships without one (each with a stated rationale), the finding kind (dependency advisory, code weakness, CI pipeline) and, for OWASP only, the CLI's OWASP 2021 category carried to 2025. When the CLI sends a finding's own CWE ids and they are a subset of its rule's, the finding maps through those alone. Every vulnerability finding (a security or secret finding with a CWE, a dependency advisory) and every CRITICAL or HIGH finding the policy covers also maps to the vulnerability-management requirement where the framework has one (SOC 2 CC7.1), which is tested by fix-by date on the policy's findings. Mapping version 2026-11-08.2. CWE™ content © The MITRE Corporation, used under the CWE Terms of Use. OWASP Top 10 © OWASP Foundation, CC BY-SA 4.0.

Requirement statuses

- Deviation noted: Vulnerability management (SOC 2 CC7.1): at least one open CRITICAL or HIGH security, secret or dependency finding is past its fix-by date, had its first scan without the finding after it in the window (when Skylos stopped detecting it, not a merge or deploy date), or is suppressed without an approval record. The counts are given, from the same findings as the policy result. This is Skylos's test of its own evidence, not an auditor's conclusion.
- Open findings: At least one open (not suppressed) finding in scope maps to this requirement.
- No open findings from mapped rules: Skylos rules map to this requirement, every in-scope project has a trusted, complete, current scan that ran those rules' checks, and none of them reported an open finding. This is not proof that the requirement is met.
- Not evidenced by static analysis: Static analysis cannot evidence this requirement (it needs process, runtime or organizational evidence), or no Skylos rule checks it. The reason is given.
- Not scanned: Skylos could evidence this requirement, but at least one in-scope project has no trusted, complete, current scan that ran the needed checks, so no claim is made.
- Scan activity recorded: The requirement is about running detection or review, and every in-scope project has a trusted, complete, current scan that ran the required checks. For vulnerability management (SOC 2 CC7.1) it also means no CRITICAL or HIGH security, secret or dependency finding is open past its fix-by date, had its first scan without the finding after it, or is suppressed without an approval record ("No deviations noted"); otherwise findings never count against it.

Policy outcomes

- Policy met, fixes due: Every in-scope project is assessed; open findings at a covered severity exist but all are still within their fix deadline.
- Policy not met: At least one open finding at a covered severity is past its fix deadline.
- Not assessed: No failure was found, but at least one in-scope project has no trusted, complete, current scan with the checks the policy needs (or nothing is in scope), so no pass is given.
- Policy met: Every in-scope project is assessed and no open finding is at a severity the policy covers.

Effort to fix

- High: a design change across components (for example adding missing authentication or authorization, replacing an unsafe serialization format).
- Low: a version, configuration value or single call changes (for example upgrading a dependency to a fixed version, pinning a CI action, turning certificate verification back on).
- Medium: code changes in one place, possibly with a credential rotation (for example parameterizing a query, replacing a weak algorithm, removing a hard-coded secret).

Terms

- Assessed project: A project whose scan is a trusted upload of the default branch (or the branch chosen for this report) that covered the whole repository, as the CLI reported it, and is no older than 90 days. Suppressions inside the scan (inline comments, Skylos config) do not stop a project being assessed; they are disclosed for that project. Neither do small gaps the CLI names in full: files it could not analyse (at most 50, and at most 2% of the files analysed) or dependencies it could not fully check (the same limits); each is a disclosed limitation.
- Before exceptions: The verdict computed as if no exception or suppression applied, so a reader can see what the exceptions change.
- Disclosed limitation: A part of the repository the scan named but did not fully analyse (files, or dependencies it could not fully check). Findings there, if any, are not in this report. The project still counts as assessed; the limitation is listed for that project and under Limitations.
- Fix required by policy: The finding is open, at a severity the policy covers (CRITICAL or HIGH security, secret or dependency finding), and must be fixed by its fix-by date for the policy to pass.
- Fix-by date: First seen date plus the deadline for the finding's severity, from the workspace's fix-deadline settings, or the Skylos defaults for CRITICAL and HIGH when none are set.
- Hidden in the scan: Findings a scan's own inline suppression comments or Skylos config kept out of its results. They are not uploaded, so the report cannot count or list them and has no approval record for them; the project's row says when this happened.
- Mapped rule: A Skylos rule whose CWE (declared by the CLI or assigned by Skylos Cloud) or finding kind falls under the requirement. Mappings are Skylos's own.
- Open finding: A finding in the selected scan that is not suppressed. Suppressed findings are listed separately with their approval record.
- Suppressed finding: A finding hidden by an approved exception, an inline suppression or a Skylos Cloud suppression.
- Trusted upload: A scan uploaded by CI with GitHub or GitLab OIDC, by Skylos Cloud itself, or with a project token an owner or admin marked for CI. Uploads with login tokens (`skylos login`) on developer machines are unverified.
- Vulnerability finding: A security or secret finding with a CWE id, a dependency advisory, or a deep scan finding with a CWE id; these map to SOC 2 CC7.1. CC7.1 tests the policy's findings (every CRITICAL and HIGH security, secret and dependency finding, with or without a CWE id) against their fix-by dates, the same population as the policy result and Skylos test SK-C2.1.

---

9. Limitations and disclaimer

- Findings and the policy result come from each project's latest assessed scan (point in time). Period evidence is the scan history, the issues no longer detected in the window (with their fix-by dates) and, in Part A, the merged pull requests with their scans, gate results and approvals; scans removed by retention are not counted. Changes pushed straight to the default branch and deployments are not evidenced by Skylos.
- Findings come from automated static analysis. Business logic, runtime configuration, deployed infrastructure and manual review are not covered.
- Dependency vulnerabilities are matched by the Skylos CLI against public advisories (OSV) when each scan ran; the advisory data snapshot date is not recorded.
- Projects that are not assessed contribute findings but no "no open findings" claims; an unverified upload can never make a requirement or the verdict look better.
- Suppressed findings are excluded from the verdict and listed with their approval record, if any. A high suppression rate weakens this evidence (NIST SP 800-53 SA-11(1)).

This report presents the results of automated static analysis by Skylos on the scope listed in the Scope section, at the time stated. Standard mappings are Skylos's own and show which findings relate to a requirement; they are not a determination of compliance, which only a qualified assessor (for example a PCI QSA, a SOC 2 or ISO/IEC 27001 auditor, or a notified body) can make. Absence of findings is not evidence of absence of vulnerabilities. Requirements marked "Not evidenced by static analysis" need manual review, runtime testing or process evidence. The full edition contains sensitive vulnerability detail; share the external summary with third parties.

10. Integrity and verification

Report ID	00000000-0000-4000-8000-000000000002
SHA-256 of the report JSON	c683c9d432c472cd2bbb350a513c75eb3f6e7799b0e61e1e3ff3d17bca3a7761
Canonical form	skylos-canonical-json/v1: UTF-8 JSON, object keys sorted by UTF-16 code units, no insignificant whitespace, members with undefined values omitted
Signature	Unsigned: no signing key was configured in Skylos Cloud when this report was generated. The SHA-256 and the verification page still detect changes to the stored report.
Verification URL	https://skylos.dev/verify/report/00000000-0000-4000-8000-000000000002
How to check a copy	The verification page says whether the record Skylos stored for this report is signed and unchanged, and shows its key facts (framework, dates, scope size, policy result and counts). A PDF, HTML or CSV copy is not itself verified: compare those facts with the copy in hand, or drop the JSON file on that page to have it hashed in your browser and compared with the stored SHA-256.
What a valid signature proves	A valid signature proves that Skylos Cloud generated exactly the stored report content (the report JSON) at the stated time, and that the stored content has not been changed since. It does not prove that the code is secure, that any requirement is met, or that a PDF, HTML or CSV copy is unchanged. Trusted scan results were vouched for by the CI provider (OIDC) or by a project token a workspace admin marked for CI; Skylos did not re-run the scans.

https://skylos.dev/verify/report/00000000-0000-4000-8000-000000000002