

SOC 2 control set for agentic teams

A lightweight SDLC for teams where AI agents write much of the code, and the SOC 2 Security controls it produces evidence for.

30

CONTROLS

18

FULLY AUTOMATED

CC1–CC9

SECURITY CRITERIA

A starting control set for a team that builds software with AI coding agents and wants to pass a SOC 2 audit without slowing down. It covers the Security criteria (the Common Criteria, CC1 to CC9). Edit it into your own. Your auditor tests the controls you write, not these.

The rule behind every control: either the pipeline enforces it and its logs are the evidence, or it's a small scheduled human job recorded as a ticket. If a control is neither, rewrite it.

The SDLC these controls assume

1. **Intent.** A human writes a short spec: the outcome, acceptance examples, and anything the change must not touch. The product owner approves it, and that approval is the authorisation. Routine dependency updates and scanner fixes can be pre-approved by a written policy.
2. **Build.** The agent designs and builds under its own identity, on an integration branch or environment. It may merge its own work there. That isn't a release.
3. **Verify.** Tests organised by the agile testing quadrants. Q1 unit tests and Q2 acceptance examples run in CI on every change. Q4 security and dependency checks run in CI and on a schedule.
4. **Release.** A human checks each spec's outcome against its acceptance examples (Q3), then approves promotion of a specific build. Only an in-scope failure or a security finding blocks a release. Exploratory testing runs outside the release, and its findings go to the backlog. The pipeline deploys with a service identity.
5. **Operate.** Monitoring, alerting, incident response, recovery. Fixes go back in at step 1.

Segregation of duties with agents

- The approval sits at **release**, not at merge, provided the merge target isn't production.
- The person who asked for a change is not the only person who approves its release.
- Agents have their own identities. They never commit or deploy as a human.
- Agents cannot change their own controls: pipeline definitions, branch policies, test configuration, protected-path lists and access settings need human code-owner review.

- An AI reviewer is a tool control, like static analysis. It is not the approver.
- The criteria allow for this. CC8.1 asks for changes to be approved "prior to implementation", and implementation is the release. CC5.1's points of focus say that where segregating duties isn't practical, management "selects and develops alternative control activities". The automated checks in this set are those alternative controls.
- Where humans do review pull requests, turn on "Require approval of the most recent reviewable push" (GitHub) or "Prohibit the most recent pusher from approving their own changes" (Azure DevOps).

The controls

Auto means a system enforces it and produces the evidence. **Sched** means a recurring human job, recorded as a ticket. Criteria references use the AICPA 2017 Trust Services Criteria numbering, checked against the TSP section 100 criteria and points of focus. The 2022 revision changed points of focus, not the criteria.

CHANGE MANAGEMENT

ID	CONTROL	KIND	EVIDENCE	CRITERIA
CM-1	Every change links to a requirement approved by the product owner, or falls under a written standing-approval policy (dependency updates, scanner fixes); merge blocked without one	Auto	Tracker link on each PR, standing-approval policy	CC8.1
CM-2	Code reaches the main branch only by pull request with passing CI	Auto	Branch protection config, PR history	CC8.1 CC6.8
CM-3	Q1 and Q2 tests run on every change and block on failure	Auto	CI run logs	CC8.1
CM-4	Production release needs approval from a human other than the requester	Auto	Environment approval log	CC8.1 CC5.1
CM-5	Only the pipeline's service identity can deploy to production	Auto	Deploy permissions, deploy log	CC8.1 CC6.1
CM-6	Pipeline, policy, test config and access config need human code-owner review	Auto	Code-owners file, PR history	CC8.1 CC5.1 CC5.2
CM-7	Agent definitions, prompts and tool config are versioned and change by PR	Auto	Repo history	CC8.1
CM-8	Emergency changes still deploy through the pipeline, on one approval, then get a full review within two working days; break-glass production access belongs to a named operations role, not developers	Sched	Exception report, review ticket, break-glass log	CC8.1

ACCESS

ID	CONTROL	KIND	EVIDENCE	CRITERIA
AC-1	Single sign-on with MFA on every in-scope system	Auto	Identity provider config	CC6.1 CC6.6
AC-2	Joiners and leavers driven from HR records; leavers removed within one working day	Auto	Identity provider audit log	CC6.2 CC6.3
AC-3	No standing admin access to production; time-limited elevation, logged	Auto	Elevation log	CC6.1 CC6.3

ID	CONTROL	KIND	EVIDENCE	CRITERIA
AC-4	Each agent has its own identity with the narrowest permissions it needs	Auto	Identity config, permission list	CC6.1 CC6.2 CC6.3
AC-5	Quarterly review of access to production, repos and agent identities	Sched	Signed-off review ticket	CC6.2 CC6.3
AC-6	Data encrypted in transit and at rest	Auto	Platform config	CC6.1 CC6.7
AC-7	Secrets held in a vault, never in code; secret scanning on every change	Auto	Scan logs	CC6.1
AC-8	Company laptops managed centrally: disk encryption, anti-malware, automatic updates	Auto	Device management reports	CC6.7 CC6.8

OPERATIONS

ID	CONTROL	KIND	EVIDENCE	CRITERIA
OP-1	Dependency and static analysis scans on every change; critical findings block	Auto	CI logs	CC7.1 CC6.8
OP-2	Infrastructure defined as code, which doubles as the asset inventory; drift detected and reported	Auto	Repo, drift reports	CC6.1 CC7.1 CC8.1
OP-3	Network boundary rules defined as code; only intended endpoints exposed	Auto	Firewall and gateway config	CC6.6
OP-4	Alerting on errors, security events and unusual access	Auto	Alert config, alert history	CC7.2
OP-5	Incidents logged, triaged and reviewed; fixes go through CM-1	Sched	Incident tickets	CC7.3 CC7.4 CC7.5
OP-6	Point-in-time backups held outside production, ideally immutable (replicas don't count); a restore tested yearly into a non-production environment	Auto + Sched	Backup log, restore test ticket	CC7.5 CC9.1

RISK, SUPPLIERS AND GOVERNANCE

ID	CONTROL	KIND	EVIDENCE	CRITERIA
GV-1	Code of conduct and leadership oversight of security, reviewed yearly	Sched	Approved document, meeting record	CC1.1 CC1.2
GV-2	Security roles written down, including a named owner for every agent	Sched	Ownership register	CC1.3 CC1.5
GV-3	Background checks on hiring where the law allows; security training on joining and every year	Sched	HR records, training records	CC1.4 CC2.2
GV-4	Security policies, including how staff report concerns, approved yearly and published	Sched	Policy repo with approval history	CC5.3 CC2.2
GV-5	System description kept current and shared with customers; a published channel for reporting security issues	Sched	Document history, contact page	CC2.3

ID	CONTROL	KIND	EVIDENCE	CRITERIA
GV-6	Yearly risk assessment, including agent and AI supplier risks	Sched	Risk register	CC3.1 to CC3.4 CC5.1 CC9.1
GV-7	Suppliers in scope, including AI model providers, reviewed yearly	Sched	Supplier register, their SOC reports	CC9.2
GV-8	Monthly review of the exception report: merges without approval, deploys outside the pipeline, emergency changes	Sched	Review ticket	CC4.1 CC4.2

Totals: 30 controls. 18 automated, 1 mixed, 11 scheduled, most of them yearly.

GV-8 matters more than its size suggests. The pipeline enforces the rules. The exception report proves you'd notice when it didn't.

WHAT THIS SET LEAVES TO YOUR CLOUD PROVIDER

Physical access (CC6.4) and disposal of physical media (CC6.5) are normally inherited from your hosting provider. Cover them through GV-7 by reviewing the provider's own SOC 2 report.

CC2.1 (using reliable information) has no single control here. It's met by the evidence itself coming from systems rather than from people filling in forms.

Optional add-on: Availability

Add this when customers want SOC 2's optional Availability criteria in scope, usually because you sell an always-on service.

High availability doesn't replace OP-6. Replication protects you when infrastructure fails, and it copies a bad migration or an accidental delete to every replica within seconds. Only a backup lets you undo that. An always-on system needs both.

ID	CONTROL	KIND	EVIDENCE	CRITERIA
AV-1	Capacity monitored, with alerts before limits are reached	Auto	Monitoring dashboards, alert history	A1.1
AV-2	Recovery targets written down (how much data you can lose, how long recovery can take); business continuity plan tested yearly	Sched	Recovery targets, test record	A1.2 A1.3
AV-3	Failover tested at least twice a year by taking down a node or zone on purpose; results recorded against the recovery targets	Sched	Test record	A1.3

Getting started

1. Build the automated controls into the pipeline first, before the first production release if you can.
2. Write the scheduled ones as recurring tickets.
3. Run a readiness review or a Type 1, and fix what it finds.
4. Open the Type 2 observation window once the controls are part of how the team works.

Questions to put to your auditor early

There's no official AICPA guidance on AI agents that I know of, so agree these before the window opens.

- Will you accept release approval as the segregation point, rather than review of every pull request?
- Do you treat the person who prompted an agent as the author of its change? (Assume yes.)
- What evidence of Q3 outcome testing do you want to see? Agree a template.
- Does an AI reviewer count for anything beyond a tool control? (Assume not.)

Sources

- AICPA, *2017 Trust Services Criteria (With Revised Points of Focus, 2022)*, TSP section 100. The authoritative criteria text. Check every mapping here against it.
- Secure Controls Framework, securecontrolsframework.com. A free control catalogue mapped to SOC 2.
- NIST SP 800-218, *Secure Software Development Framework*.
- Lisa Crispin and Janet Gregory, *Agile Testing and More Agile Testing*. The testing quadrants.
- Microsoft Learn, Azure Repos branch policies. GitHub Docs, protected branches.