

GUIDE · FREE TO READ

Surviving the enterprise security questionnaire

The contract was two weeks from signature. Then it went to their security team, and back came a spreadsheet with a hundred and eighty rows, half of which assume you employ someone whose whole job is answering them. You do not. This is what the rows are actually for, which of them you can answer today, and the two answers that lose the deal outright.

No email required, nothing gated. Written for a founder with no security team, no SOC 2, and a quarter they cannot afford to lose.

Contents

- 1 The questionnaire is a proxy, and not for your security
- 2 Your deal is this quarter; the report is next year
- 3 Most of the spreadsheet does not need a certificate
- 4 A delivery record lets you answer from evidence
- 5 Two answers that end deals
- 6 What we do not hold
- 7 What to do on Monday

1 The questionnaire is a proxy, and not for your

security

The first thing to understand is that the person reading your answers is not trying to work out whether you are secure. They cannot. Nobody can establish that from a spreadsheet, and they know it better than you do. What they are working out is whether they can explain the decision to buy you — to their auditor, to their risk committee, and to whoever asks the question after an incident.

That changes what a good answer looks like. A reviewer can forward a document. They cannot forward your confidence. An answer that is true but rests only on your word gives them nothing to put in a file, so it scores worse than a modest answer with an artefact attached — even when the modest answer describes a weaker control.

Read every row as: if this vendor causes an incident, what will I be able to show that I checked?

This is also why the process takes as long as it does, and why the cost of failing it is a deal rather than a discount:

- **36%** of companies lost a deal because they lacked a required certification — up from 29% the year before — [SOC 2 industry survey, 2026](https://soc2auditors.org/insights/soc-2-compliance-for-startups/) (<https://soc2auditors.org/insights/soc-2-compliance-for-startups/>)
- **4–6 weeks** for a typical vendor security assessment, once you have supplied everything asked for — [Vendor assessment benchmarks, 2026](https://getagency.com/vendor-security-review/) (<https://getagency.com/vendor-security-review/>)
- **10–11 months** the common path to a SOC 2 Type 2 report: Type 1 at month four, then a six-month observation window — [SOC 2 timing benchmarks, 2026](https://hub.causo.ai/guides/soc2-for-seed-startups-enterprise-deals-2026) (<https://hub.causo.ai/guides/soc2-for-seed-startups-enterprise-deals-2026>)

Published industry figures, directional rather than audited. Treat them as the shape of the problem, not as a forecast of your deal.

Note what the first figure is measuring. It is not that insecure vendors lost deals. It is that vendors without the paperwork lost deals, and the share doing so is rising. The review is a documentation exercise wearing a security title, and the good news buried in that is that documentation is something a small team can actually produce this month.

2 Your deal is this quarter; the report is next year

Then you look up what SOC 2 Type 2 takes. Roughly four months to the Type 1 report, then a six-month window during which an auditor observes that your controls actually operate, then the Type 2 report itself. Ten to eleven months, on the common path, for something your buyer wants before the quarter closes.

Founders respond to that gap in one of two bad ways. They stall, hoping the buyer relents, and the deal quietly ages out. Or they write something on the website with the word “compliant” in it and hope nobody asks for the report. Section 5 is about why the second one is worse than losing the deal.

SOC 2 is not the instrument that saves this deal. It is the instrument that saves the next four. Start it, and separately go and win this one.

Four things to do in the meantime, in this order:

1. **Ask for the actual control list.** Much of what arrives is the reviewer’s default template, not a contractual requirement. Asking which rows are mandatory for a vendor at your tier and your data scope frequently removes a third of them, and it takes one email.
2. **Ask what the exception path is.** Nearly every enterprise vendor programme has one, usually a conditional approval with a remediation date and a review. It exists because the programme would otherwise be unable to buy from anyone under fifty people. Nobody will offer it to you unprompted; asking is not a sign of weakness, it is a sign you have read their process.
3. **Send a dated position, not a promise.** One page: what is true today, what is not yet true, and the date each gap closes, signed by a named person. This is the artefact the reviewer has been trying to extract from you all along, because it is the one they can attach to the file.
4. **Start the clock anyway, today.** The observation window only begins when it begins. Every week you spend deciding whether to start is a week added to the end, and the next buyer will ask the same question.

3 Most of the spreadsheet does not need a certificate

Strip out the rows that genuinely require an external auditor and what is left is mostly four things. None of them need a report, and all of them are within reach of a team of a dozen people.

A documented process

A page and a half, not a manual. Named steps, what has to pass before a release, who is allowed to approve one, and what happens when something is rejected. Written down and actually followed beats comprehensive and aspirational, because the audit tests the second one and finds the difference in about an hour.

Evidence attached to each change

For any change the reviewer picks, you should be able to produce the ticket, the checks that ran, what each returned, and who approved the merge. If you can do that for a change they choose rather than one you choose, you have answered a large block of the spreadsheet in a single attachment.

A dependency and licence position

An inventory of everything you ship that you did not write: the versions, the licences, the known vulnerabilities, and the date the list was generated. This is the row most often missing and the cheapest one to fix — the tooling is free and the list takes an afternoon the first time.

Incident and access practice

Who can reach production, how that access is granted and removed, when the list was last reviewed, and what happens at two in the morning. One real incident written up with times — detected, contained, communicated, closed — is worth more to a reviewer than a policy nobody has ever exercised, because it proves the process survived contact with an actual event.

The licence half of the third item is worth its own sentence, because it tends to be treated as a lawyer's problem and then discovered by an engineer:

- **68%** of audited codebases contain open-source licence conflicts — Technical due diligence benchmarks, 2026 (<https://hypernestlabs.com/guides/technical-due-diligence-checklist>)

Published benchmark, directional rather than audited. It is a diligence figure rather than a security one, and it turns up in both reviews for the same reason: nobody generated the list.

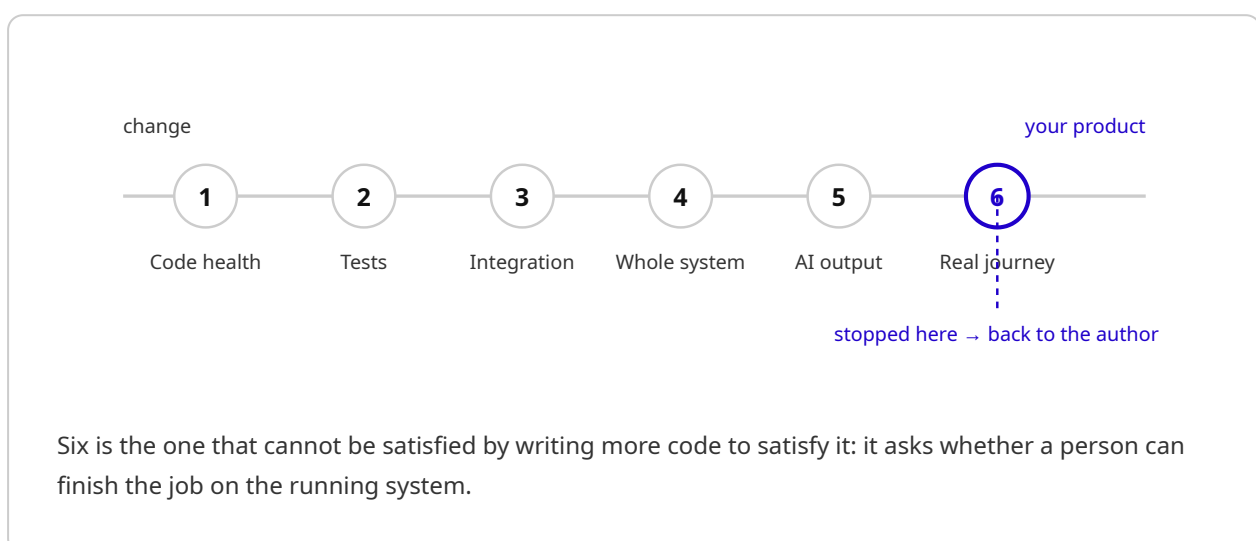
4 A delivery record lets you answer from evidence

Almost every row in a security questionnaire is asking about a property of how you build software. There are only two ways to answer that kind of question. You assert the property, which costs a minute and buys exactly as much trust as the reviewer already had in you. Or you show it, which costs whatever it cost to have kept the record.

That is the whole argument for keeping one. If the record is a by-product of shipping — every item tracked, every check logged with its result, every decision written down where it was made — then it costs nothing at questionnaire time, because it already exists. If it is not, you are reconstructing six months of history under deadline while the deal ages, and the reconstruction is itself the thing an auditor will not accept.

What the row asks	What you attach instead of an assurance
Do you have a documented development lifecycle?	The written process: named steps, what blocks a release, who may approve one.
Is every change reviewed and tested before production?	For a change the reviewer picks at random: the ticket, the checks that ran, each result, and the approver.
How do you manage vulnerabilities in third-party code?	A dated inventory of what you ship, its versions and its licences, plus the scan output from the last build.
Do you have change management and segregation of duties?	Who opened the work, who approved the merge, and the record showing they were different people.
Who has access to production, and how is it removed?	The current access list, the joiner and leaver steps, and the date of the last review.
What is your incident process?	The runbook, and one real incident written up with times: detected, contained, communicated, closed.

For context on what “the checks that ran” can mean, our own process puts six of them in front of every change, and the record of which passed is per change rather than per release:



- **Security and code health** — Insecure code, leaked credentials and unreadable work stopped at the door. (every change)

- **Automated tests** — Every change proved against the test suite — and the bar rises as your product grows. (every change)
- **Integration** — Confirms the parts still work together, not just on their own. (on integration)
- **Whole system** — Rebuilds everything from scratch and scans the full stack for vulnerabilities. (on integration)
- **AI output quality** — What the AI produced is sampled and scored. Nothing is assumed. (before release)
- **Real user journey** — Can a real person finish the job, end to end, on the running system. (before release)

You do not need six, and you certainly do not need ours. You need the answer to “what ran, what did it say, and can you show me for this particular change” to be a lookup rather than an investigation.

A record that only produces good news is not evidence. It is marketing with timestamps, and reviewers have seen a lot of it.

Ours reports that 43% of work was later redone, alongside the figures that flatter us, on the position dated 2 August 2026. That number is embarrassing and it is the reason the rest of the board is worth reading: a reviewer who finds one uncomfortable figure in your evidence trusts the others more, not less. If you are assembling a pack this month, leave the awkward row in.

One limit, stated plainly. A delivery record does not substitute for a certification where the buyer’s policy names one. If their procurement standard says the vendor holds the report, no amount of evidence changes that, and the honest move is to find out in week one rather than week six. Ask directly: is this a hard gate, or is it a control the review is trying to satisfy some other way?

5 Two answers that end deals

Everything above is about scoring badly and recovering. These two are different in kind. They do not cost you points; they cost you the buyer’s ability to defend having chosen you, which is the only thing keeping the deal alive.

Never claim a certification you do not hold.

This includes the softened versions, which are the ones founders actually reach for: describing an audit you have started as though it has concluded, saying you are “aligned with” or “compliant with” a standard you have not been assessed against, or putting a badge on a website in the belief that nobody reads the small print. Someone does. The request that follows is always the same — please send the report — and there isn’t one. At that point the conversation stops being about security. Your buyer’s security lead now has a vendor who made a false statement in writing during procurement, and no remediation plan fixes that.

Never answer “yes” to a control you have not implemented.

Questionnaire answers are frequently pulled into the contract as representations, which means a hopeful yes is a warranty. It gets discovered in one of two places: at the audit, where a tester asks for the evidence behind the answer, or during an incident, where the answer becomes the first thing anyone reads. The second is worse than the first by a wide margin.

The replacement is not a clever form of words. It is: *no, here is what we do instead, and here is when that changes*. Reviewers accept that far more often than founders expect, because a documented no with a date is something they can process, and a yes they cannot verify is something they have to carry personally.

A practical rule that prevents most of this: whoever fills in a row must be the person who would have to produce the artefact for it. When the questionnaire is answered by whoever is closest to the commission, every ambiguous row resolves optimistically, and the optimism is discovered later by someone with more authority than you.

6 What we do not hold

We should be direct about our own position, because this page would be worthless coming from a supplier who was vague about it.

NamiQ does not hold SOC 2. We have not been audited against it and we are not in a Type 2 observation window. Nothing on this site says otherwise, and if you ever find something that reads as though it does, it is wrong and we want to know.

We are also not auditors, and we do not issue attestations of any kind. What we do is build software with the evidence trail attached, which happens to be most of what a vendor review asks you to produce. That is a useful thing and a narrower thing than “we get you

certified”, and any supplier blurring the two is worth a harder question. Apply the same test to us that this guide tells you to apply to a questionnaire answer: ask what we would attach.

7 What to do on Monday

Cheapest first. The first three cost a day between them and collectively move more rows than anything else on the list.

1. **Email the reviewer and ask two questions.** Which rows are mandatory at our tier and data scope, and what is your exception path. Free, and it regularly removes more work than the rest of this list adds.
2. **Generate a dated dependency and licence inventory.** An hour with tooling you already have. It is the most commonly missing artefact and one of the easiest to produce.
3. **Write the process down in a page and a half.** Describe what you actually do, including the parts you are not proud of. Aspirational documents fail audits; accurate ones pass with findings.
4. **Pick one change from last week and assemble its evidence end to end.** Ticket, checks, results, approver. If you cannot, you have found the gap in your own time instead of theirs — which is the entire point of doing it on a Monday.
5. **List who can reach production and remove whoever should not.** Then write the date of the review at the top of the list, because the date is half of what is being asked for.
6. **Write the incident runbook and rehearse it once.** An hour. A rehearsed process with a written record beats an elegant one nobody has run.
7. **Answer the spreadsheet honestly, with dates against every no.** Send the one-page position alongside it rather than after it.
8. **Start the SOC 2 clock, and tell the buyer the date.** A named start date does work for you immediately, months before the report exists.

None of this needs our tooling, and most of it needs no tooling at all. The part that is genuinely hard to retrofit is the fourth item — being able to produce the full evidence for a change somebody else picked. If you would rather that were a by-product of how the work ships than a project of its own, [that is what the delivery process does](#). And if the eight steps

above get you through the review without speaking to us, that is a good outcome and the reason this is not behind a form.