PLAYBOOK · V1 · 2026

The 90-Day AI Startup Security Playbook

How pre-Series-A AI founders unlock enterprise pipeline through SOC 2, fractional security leadership, and an AI-native four-layer stack

By Jared Smith · sublimecoding.com

This playbook is for one specific reader: a pre-Series-A AI startup founder or CTO with a named enterprise prospect stuck at the security questionnaire. Most of you arrived here because you've already read three or four of the underlying essays and now want them as one document — to send to the board, to a co-founder, to the head of sales, or to read end-to-end on a flight.

The playbook compiles four flagship essays into a single 90-day argument:

- **SOC 2 is a revenue tool, not a security tool.** Reframe the audit as a sales project to make the math work.
- **vCISO math.** Why fractional security leadership at $2–4K/month beats a $300K full-time CISO at this stage.
- **The AI-native four-layer stack.** Prompt injection, agent credentials, secrets, audit logging — the layers classic appsec doesn't cover.
- **What 90 days of a fractional engagement actually looks like.** Week-by-week deliverables, real costs, and what graduation looks like.

Each chapter is an essay that's been published on sublimecoding.com (https://sublimecoding.com/blog). The web versions get edited and updated; this PDF is a snapshot. When the math changes, the source posts change too.

How to use this playbook: read the cover-to-cover argument if you're starting from scratch. Skip to a single chapter if you're already deep on one decision and want clarity on another. Send the PDF link to the person at your company who needs to hear the case made — co-founder, board member, head of sales — without having to convince them to read four blog posts.

Table of contents

1. Chapter 1 — SOC 2 is a Revenue Tool, Not a Security Tool
2. Chapter 2 — vCISO Math for AI Founders
3. Chapter 3 — How I'd Run Security at an AI-Native Company in 2026
4. Chapter 4 — What 90 Days of a Fractional Security Engagement Actually Looks Like
5. How to use this playbook from here

CHAPTER 1

Chapter 1 — SOC 2 is a Revenue Tool, Not a Security Tool

> Start here. The single most expensive misconception about SOC 2 at pre-Series-A AI startups is that it's a security project. It isn't — it's a sales project, and that reframing changes everything about how you scope it, how long it takes, what it costs, and which engineer ends up running it. This chapter makes the case and gives you the 90-day playbook.

SOC 2 is a revenue tool, not a security tool.

KEY TAKEAWAYS

- **Total cost is $25–45K over 90 days for a 15-person AI startup.** Auditor fees, tooling (Vanta or Drata), and operator time — not "burn six months and your best engineer."
- **Scope it as a sales project, not a security project.** Pick the controls that unlock the named enterprise prospect waiting on the report; defer the rest.
- **Days 1–30 stop the bleeding; 30–60 close the gaps; 60–90 get the report.** Anything outside that path is yak-shaving.

Every AI founder pre-Series A gets this wrong. You scope the audit like a security project and hand it to your best engineer. Six months later you've burned your strongest IC, the report still isn't done, and the enterprise deal you were trying to close went to a competitor with the checkbox.

Reframe it.

Your engineering team already thinks about auth, secrets, and data handling harder than any auditor will. SOC 2 doesn't make you secure. It unlocks the pipeline you're already leaving on the table. The VP of Engineering at that Fortune 500 who loves your demo cannot send you a contract without it.

So stop scoping it as a security project. Scope it as a sales project. And run it in 90 days.

Here's the path I've used at AI startups:

Days 1 to 30: stop the bleeding

- Pick a compliance platform (Vanta, Drata, Secureframe). Don't overthink it.
- Name one internal DRI. Not a committee. One person owns it end to end.
- Target Type I first. Type II comes after you've operated controls for 6 months.

- Retain a vCISO for 5 hours a month. $2 to 4k. Worth every dollar.
- Pull policies off the shelf from the platform. Don't write your own. Most platforms have this built in. Some are better than others.

Days 30 to 60: close the gaps

- MDM on every laptop. Non-negotiable.
- SSO and MFA across every tool, including the cheap ones nobody wants to pay to upgrade.
- Background checks on employees. 48 hours.
- Vendor review process. A spreadsheet is fine for now.
- Logging and quarterly access reviews. Most startups skip these. Auditors don't.

Days 60 to 90: get the report

- Book the audit with a reputable firm. Don't pick the cheapest.
- Run a mock audit with your vCISO two weeks before kickoff.
- Fix the 10 things they find. There will be 10.
- Get the Type I report in hand.
- This can take longer than 30 days depending on how responsive the team is to issues.

Cost and timeline

Total cost for a 15-person startup: usually $25k to $45k all in.
Timeline from kickoff to report: 90 to 120 days if you're serious.

What it unlocks

Every enterprise deal stalled at "send us your SOC 2" moves to contract. This can turn theoretical hundreds of thousands and in some cases millions in ARR pipeline into closed revenue inside a quarter.

The mistake most founders make is treating the audit itself as the security work. It isn't. The audit is the door opener. The real security work starts after, once you're actually operating the controls day to day and your customer success team stops losing deals to a missing PDF.

If you're pre-Series A, AI-native, and watching enterprise deals die at the security review stage, this is the lever.

Read this next

- **vCISO Math for AI Founders: Why 5 Hours a Month Beats a Full-Time Hire** — If you're scoping it as a sales project, this is who you hire to run it.

- **How I'd Run Security at an AI-Native Company in 2026** — What the controls behind the audit actually look like at an AI-native company.
- **How We Cut $350K From Cloud Spend in 6 Months (And What I'd Do Differently)** — Same playbook framing applied to your cloud bill — treat it as a contract, not an architecture problem.

Canonical web version: sublimecoding.com/blog/soc-2-is-a-revenue-tool-not-a-security-tool

CHAPTER 2

Chapter 2 — vCISO Math for AI Founders

> Once you've decided to run the SOC 2 process, the next decision is who runs it on your side. Most founders default to either "I'll do it" (wrong — opportunity cost too high) or "we need a CISO" (wrong — over-leveled at this stage). The answer is fractional security leadership. This chapter walks through the math: what a vCISO costs, what they actually do, and when you graduate to a full-time hire.

Don't hire a CISO. Rent one.

KEY TAKEAWAYS

- **Full-time CISO costs $200–400K all-in; fractional vCISO costs $2–4K/month.** Most pre-Series-A startups need 5 hours of strategic security work per month — not a full-time executive.
- **What a vCISO actually does is policy, vendor security review, audit prep, and architecture review — not detection-and-response.** If you need a SOC, hire one separately.
- **Graduate to a full-time CISO when security work consistently exceeds 20 hrs/week, you're past Series A with 50+ engineers, or a regulated deal requires the org-chart line.** Before then, full-time is over-leveled.

This is the single most actionable security advice I give to pre-Series-A founders, and the one most consistently ignored. The pattern is predictable: an enterprise prospect asks for a SOC 2 report, the founder panics, posts a security-leadership job opening with a $250K base, sources for two months, and either hires the wrong person or gives up and ships the report without leadership in place. Both outcomes are bad. Both are avoidable.

The right answer at this stage is a fractional vCISO. Five hours a month, $2–4K, retained on a recurring contract. Below, the math, what to expect, and when to graduate to a full-time hire.

The math

Let's compare the two paths concretely.

Full-time CISO at a 15-person AI startup, pre-Series-A:

- Base salary: $220–320K (Bay Area / NYC / remote-but-competitive)
- Equity: 0.5–1.5% (roughly $50–200K paper value at this stage)
- Benefits and overhead: ~25% of base = $55–80K

- Recruiter fee (if external): 20–25% of first-year comp = $50–80K one-time
- **First-year cash cost: $325–480K. Year-two onward: $275–400K.**

Fractional vCISO, 5 hours a month:

- Hourly: $400–800 depending on market and experience level
- Monthly: $2–4K
- Annual: $24–48K
- No equity, no benefits, no recruiter fee
- **First-year cash cost: $24–48K. Same year-two.**

The vCISO is roughly 5–10% of the cash cost of a full-time CISO and zero equity. For a pre-Series-A company where every dollar is runway, this is the difference between two months and twenty months of additional runway tied up in the security function.

You might object: "But a full-time CISO does much more than 5 hours a week." That's true. They do roughly 160 hours a month. The question is: does your 15-person AI startup, which has zero customers in regulated industries, has not yet had a security incident, and is six months from its first SOC 2 audit — does it actually have 160 hours a month of CISO-level work to do?

It does not. It has roughly 5–20 hours a month of CISO-level work, plus a much larger volume of engineering-led security execution that the engineering team is already doing or should be doing. A vCISO sized to the actual volume of CISO-shaped work is the right tool.

What a vCISO actually does (and what they don't)

The biggest source of disappointment with vCISOs is mismatched expectations. Here's what to expect for $2–4K a month.

What they do:

- **Strategic guidance.** Quarterly review of your security roadmap, threat landscape, and gaps relative to your customer base. They tell you what to worry about and in what order.
- **Audit and certification readiness.** They read your evidence, tell you what's missing, and prep you for the auditor's conversation. Most vCISOs have shepherded ten to fifty SOC 2 audits and know exactly which controls auditors actually scrutinize.
- **Customer security questionnaires.** Enterprise prospects send 80–200 question security questionnaires. Your vCISO either fills them out or directs your team on the answers. This alone usually pays for the engagement.
- **Incident-response support.** When something goes sideways, they're on the phone in two hours. They've handled incidents before. Your engineering team has not.

- **Policy authorship and review.** Information security policy, acceptable use policy, vendor risk policy, incident response plan. They have templates. They customize them. They sign them. Done in days, not weeks.
- **Auditor relationship.** A reputable vCISO has working relationships with multiple audit firms. Their warm intro to a CPA firm gets you a faster engagement and a better rate.

What they don't do:

- Hands-on engineering. They don't write code, configure SSO, or set up MDM. Your engineering team does that under their guidance.
- 24/7 monitoring. They are not your SOC. If you need real-time monitoring, you're hiring an MSSP, not a vCISO.
- Hire and manage a security team. They might help you scope the first hire when you're ready, but they're not running people.
- Live in your Slack. Five hours a month is five hours a month. They will not be available for ad-hoc questions multiple times a day.

Match your expectations to the contract and the relationship is wildly productive. Mismatch and you'll fire each other within four months.

When to graduate to a full-time hire

The vCISO model has a ceiling. The signals that you've hit it:

1. **You're spending 20+ hours a month on the engagement.** If you've stretched a 5-hour retainer into 20 hours of effective work, you're paying overage rates and the vCISO is bottlenecked. Time to bring it in-house.
2. **Your security team is more than 2 people.** A vCISO can guide one or two security ICs. Beyond that, you need a security leader with capacity to actually manage.
3. **You're regulated.** If you take on PCI Level 1, HIPAA covered-entity status, FedRAMP, or financial services charters, the regulator's expectation of a named, in-house CISO becomes binding. Hire.
4. **You're past Series B and selling to F500 enterprises.** At that revenue scale your customer expectations include a real CISO they can put on the phone. The vCISO can no longer carry that representational load.
5. **You've had a security incident that drew a board-level response.** Boards want a named accountable person. Don't argue with that.

Pre-Series A: vCISO. Series A through B: vCISO with the option to upgrade. Series B+: full-time, almost always.

The bad-vCISO red flags

Not all vCISOs are equal. Five flags I've learned to watch for:

1. **They've never been an in-house security leader.** Career consultants who've never had to actually live with their decisions tend to over-prescribe. Look for someone who's been a Director or VP of Security at one or more real companies and decided to go fractional.
2. **They don't ask about your customers.** If the vCISO doesn't immediately want to know who buys from you and what their security expectations are, they're going to give you generic advice. Your security program should be shaped by the people writing the checks, not by a checklist.
3. **They sell products.** Some "vCISO" engagements are thinly disguised channel partnerships for compliance platforms or security tooling. They'll push you toward whatever they get paid to push. Ask up front: do you have any reseller, referral, or affiliate relationships with the platforms you'll recommend?
4. **They quote you "all-in flat-rate" pricing.** The honest pricing is hourly with a monthly retainer minimum. Flat-rate vCISO pricing for $1,500 a month usually means you'll get attention only when you complain.
5. **They can't name three audit firms they'd recommend.** A real vCISO has done a lot of audits and has opinions about who's good and who's bad. If they shrug at this question, they haven't done the volume.

How to interview a vCISO in 30 minutes

A short list of questions that surface signal fast:

- "What's the right SOC 2 audit firm for a 15-person AI startup?" — They should name two or three with rate ranges and tradeoffs.
- "What are the three controls auditors most often flag at a company our size?" — They should answer in 30 seconds without thinking. Common answers: access reviews, vendor management, change management documentation.
- "Walk me through the last incident you led." — Listen for structure. Did they have a runbook? Who was in the room? What was the post-mortem? Vague answers are a flag.
- "What would you tell my engineering team to start doing on Monday?" — They should have a concrete short list. If it's "depends on a deeper assessment," they're billing for the assessment.
- "What gets you fired?" — Good answer: "I get fired when the auditor finds things I should have flagged in advance, or when I told you something was fine and it wasn't." Bad answer: long pause.

The deliverables to write into the contract

Don't sign a vCISO contract without specifying outcomes. Generic monthly retainers float into nothing. Concrete examples:

- SOC 2 Type I readiness in 90 days
- Information security policy + 4 supporting policies signed and ratified within 30 days
- Quarterly risk register reviewed and updated
- Customer security questionnaires turned around in 5 business days
- Incident-response participation within 4 hours of declared incident, any time
- Quarterly readout to founders / board with current posture and gap list

If they push back on writing these into the contract, they're not committing. Find a different vCISO.

The honest tradeoffs

To be fair to the full-time CISO model: there are real things you give up by going fractional.

You don't get a leader who's in your Slack every day, building relationships with engineers, customers, and the board over a sustained period. The institutional knowledge of an in-house leader compounds — they know which engineer cuts corners, which customer is going to ask which question, which board member wants which level of detail. A vCISO will never have that depth.

You also lose the recruiting halo. A named, in-house CISO with a strong reputation can be a meaningful asset when you're hiring senior security engineers or selling to security-sensitive customers. The vCISO does not show up on your team page.

And you lose the optionality of having someone in seat when things go sideways. If you have an incident on a Saturday, your full-time CISO is on it. Your vCISO is on it within a few hours, but those hours can matter.

The honest framing: the vCISO model trades depth-of-context for cost efficiency. At fifteen people pre-Series-A, the cost efficiency wins by a wide margin. The depth-of-context cost is small because there's not yet much context to be deep about. As the company grows, that math flips, and you should flip with it.

The takeaway

Your security program at 15 people, pre-Series-A, looks like:

- Engineering does the engineering security work (auth, secrets, IAM, deployment hygiene). They were doing this anyway and are better at it than any external person.

- A vCISO does the leadership, audit, and customer-facing security work. Five hours a month, $2–4K, deliverables in the contract.
- Your founder owns the customer-facing risk story until the company outgrows them.

This setup costs you $24–48K a year and 5% of the leadership burn of a full-time CISO. It unlocks SOC 2, Vendor Risk Assessments, and enterprise customer questionnaires — the unlocks that actually move revenue. And when you outgrow it, around Series B, you graduate to a full-time hire with a much clearer view of what good looks like, because you've been working with one for two years.

The mistake is treating the security leadership question as a binary "no one" or "full-time hire" problem. There's a perfectly engineered middle option, and it's the right one for the first three years of an AI-native company's life. Use it.

Read this next

- **SOC 2 Is a Revenue Tool, Not a Security Tool** — What you ship in 90 days once you've hired the vCISO.
- **How I'd Run Security at an AI-Native Company in 2026** — The technical security stack the vCISO will help you build.

Canonical web version: sublimecoding.com/blog/vciso-math-for-ai-founders

CHAPTER 3

Chapter 3 — How I'd Run Security at an AI-Native Company in 2026

> SOC 2 covers classic appsec controls. It does not cover the four layers that actually break in AI-native companies: prompt injection, agent credential scoping, secrets handling, and audit logging. This chapter is the technical playbook for the AI-specific work that runs in parallel to your SOC 2 audit — and the 90-day plan for standing it up.

AI-native companies need a security model that classic appsec doesn't cover. Most don't have one.

KEY TAKEAWAYS

- **The four-layer stack: prompt injection defense, agent credential scoping, secrets handling, audit logging.** None of these come for free in classic appsec.
- **Start with credential scoping in the 90-day plan.** That's where the largest blast-radius incidents originate; everything else builds on top.
- **Defer SIEM, formal threat modeling, and a full bug bounty program until post-Series-A.** Pre-A, you pick the high-leverage layers; the rest is operational debt.

The pattern I see across early-stage AI companies: a strong engineering team treats security like a 2018 SaaS product — auth, secrets, the SOC 2 checklist. Meanwhile their product is shipping autonomous agents with cloud credentials, accepting unstructured input from customers as the primary interface, and training models on data the customers didn't fully realize they were exposing. The threat model has changed. The controls haven't kept up.

If I were building the security program at an AI-native company today, this is the layered stack I'd put in place, the things I'd ship in the first 90 days, and the things I'd consciously defer.

The four-layer stack

Classic appsec is one layer of four. Treating it as the whole picture is the most common mistake I see.

Layer 1 — Classic application security

This is everything that's been good practice for fifteen years and doesn't go away because you're an AI company. Auth and authorization. Secrets management. Input validation. SQL injection prevention. CSRF tokens. SSRF guardrails. TLS everywhere. Least-privilege IAM. Logging and audit trails. Backups and recovery.

This layer is solved. The advice has been written down a hundred times. If you're not doing it, do it. If you are, skip the rest of this layer's discussion and move on. The interesting work for AI companies is in the next three layers.

Layer 2 — Data security and the training question

The novel question for AI-native companies is what data goes into the model and where it ends up.

The threats:

- **Training-data exfiltration.** A model trained or fine-tuned on customer data can leak fragments of that data through generation. This is real, has been demonstrated repeatedly, and is not solved by "we delete the data after training."
- **Prompt-context leakage.** Customer A's data ends up in customer B's response because both customers share the same backend prompt context. RAG pipelines are the worst offender here.
- **Vendor-side training.** You send customer data to a foundation model API. The vendor uses it to improve their model. Your customer didn't consent to that.

The controls I'd ship:

- **Tenant isolation in retrieval.** Every vector-DB query and every RAG retrieval must filter by tenant ID at the index level, not in post-processing. This is the single most common AI company security bug I see in code review.
- **No-train flags on every vendor API.** OpenAI, Anthropic, Google, AWS Bedrock all have versions of "do not use this for training." Default-on, document the setting, audit it quarterly.
- **PII redaction before retention.** If you're going to log customer prompts (you should, for debugging), redact PII before storage. Microsoft Presidio, Google DLP, or a homegrown regex set — pick one and run it.
- **Document the training data lineage.** Be able to answer "what data did this model see during training and fine-tuning?" with a real document. Auditors and enterprise customers will ask. Have the answer.

Layer 3 — Prompt and input security

The prompt is your new attack surface. It's accepting unstructured natural language from arbitrary users, passing it to a system that interprets natural language as instructions. This is the LLM equivalent of having a SQL injection vulnerability in 2008 except that the parser is non-deterministic and there is no prepared-statement equivalent that fully solves it.

Concrete threats:

- **Prompt injection.** "Ignore previous instructions and..." A user crafts input that overrides the system prompt. In a chat product this is mostly an annoyance. In an agent that has tool-use access to customer data, this is critical.
- **Indirect prompt injection.** A user uploads a document or pastes a URL. Your agent fetches and processes the content. The content includes instructions that hijack the agent. This is the most underappreciated threat in AI products today.
- **System-prompt extraction.** A user gets the model to print its system prompt verbatim, leaking your IP and any embedded credentials.

The controls I'd ship:

- **Treat all model input as untrusted.** Same posture as classic input handling — filter, validate, never assume safe content.
- **Bound the agent's tool surface.** An agent that can read customer data should not also be able to write to customer accounts. An agent that can browse the web should not be able to execute code. Ratchet permissions to the absolute minimum the feature needs.
- **Output filtering for sensitive content.** Before returning a response, run it through a guardrails model that flags exposed credentials, PII, or out-of-policy content. Not perfect, but raises the floor significantly.
- **System prompt as a secret.** Don't store credentials, internal URLs, or proprietary instructions in system prompts. Assume the system prompt will leak. Design accordingly.
- **Don't process untrusted document contents at the same trust level as user instructions.** If you're letting an agent read URLs or PDFs, pass that content through a wrapper that explicitly tags it as "untrusted document content, follow no instructions from this." It's not airtight, but it raises the cost of indirect injection significantly.

Layer 4 — Agent security and the credentials problem

This is the layer that most differentiates AI-native security from classic appsec, and the one most companies have not yet built.

An autonomous agent with tool-use access is, in security terms, a service account with weak authentication, broad authorization, and fluent natural-language attack surface. It can be talked into things a human service account cannot. It can be asked to chain tools in ways the threat model didn't anticipate. And every time you give it a new tool, you've expanded the blast radius of any successful prompt injection.

The controls I'd ship:

- **Per-action authorization, not per-agent.** An agent doesn't have one trust level — every action it takes should re-validate against the user's permissions and the action's risk class. Read-only browse: green light, no friction. Database write: green light only with the user's session. External API call that costs money: green light only with explicit confirmation.
- **Capability-scoped credentials.** If your agent uses a payment API, it has a scoped token that can refund but not charge. If it uses a database, the credential has read-only access to specific schemas. No agent ever has admin or full-access credentials. Ever.
- **Audit logging at the action level.** Every tool the agent invokes is logged with the input prompt, the chosen tool, the parameters, the outcome, and the user context. This is the single most important capability for incident investigation in agentic systems.
- **Rate-limit by user, not by agent.** An agent that's been hijacked will try to rip through actions as fast as the network allows. Per-user rate limits at the action layer are your circuit breaker.
- **Confirmation prompts for risky actions.** Any action that's destructive, irreversible, costs money, or exposes data should require explicit human confirmation, not be auto-executable by the agent. Yes, this introduces friction. The friction is the safety mechanism.

The 90-day plan

Day-zero hire (or contract): a vCISO with AI-native experience. Don't try to build this without one. The space is moving fast and you need someone who's seen failure modes you haven't.

Days 1–30: foundations.

- Layer 1 baseline: SSO, MFA, MDM, secrets vault, IAM least-privilege review.
- Layer 2 controls: no-train flags everywhere, RAG tenant isolation audit.
- Set up audit logging at the action level for any agent or tool-using LLM.
- Document model lineage for every model you ship.

Days 30–60: prompt and agent.

- Adversarial review of every system prompt. Assume it will be extracted; remove anything that should not be public.
- Tool-permission audit: every agent's available tools, mapped to risk class, with confirmation gates added where missing.

- Indirect-prompt-injection testing on document and URL ingestion paths.
- PII redaction in logs and analytics pipelines.

Days 60–90: program.

- SOC 2 Type I readiness, scoped to include AI-specific controls (data lineage, no-train, agent action logging). Most off-the-shelf SOC 2 templates do not include these.
- Customer-facing security documentation: trust page, AI usage disclosure, data handling policy. Enterprise prospects will ask.
- Incident response runbook with AI-specific scenarios: prompt injection at scale, data exfil via training, agent runaway.
- Quarterly security review cadence with founders and key engineering leads.

What I'd defer

The instinct in security programs is to over-include. At the speed an AI startup moves, that's fatal — every control has a maintenance cost, and a security program that pisses off engineering will be worked around inside a quarter.

Things I'd consciously defer at the early stage:

- **Heavy DLP tooling.** Worth it at scale, distracting at fifteen people.
- **Endpoint detection and response.** MDM gets you most of the value at this stage. Real EDR comes after Series B.
- **SIEM platforms.** Centralized logging is great. A full SIEM with detection rules is overkill before you have a security team to run it.
- **Bug bounty programs.** Run them once you have a triage process. Before that, they generate noise.
- **Penetration tests beyond what your customers require.** One annual pentest scoped to your customer requirements is enough until you're in a regulated vertical.

The discipline is doing the controls that matter at your stage and not the ones that look impressive on a security marketing page.

The takeaway

AI-native security is not classic appsec plus "be careful with prompts." It's a four-layer stack, and three of those layers — data, prompt, agent — are mostly novel relative to where most engineering teams have built up muscle memory.

You will get most of the value from **tenant isolation in retrieval, scoped credentials for agents, action-level audit logging, and confirmation gates on destructive actions.** Those four controls handle the vast majority of the AI-specific failure modes I've seen at production scale.

Everything else is sequencing and discipline. Don't skip Layer 1. Don't pretend Layers 2–4 don't exist. Hire a vCISO who's seen this space before. Document what you do and don't do, because your customers, your auditors, and your future self will all want to know.

The companies that get this right in the next two years will look like reasonable enterprise vendors. The ones that don't will spend a quarter on incident response that should have been spent on product.

Read this next

- **SOC 2 Is a Revenue Tool, Not a Security Tool** — How to convert this security posture into the audit report your enterprise pipeline is asking for.
- **vCISO Math for AI Founders: Why 5 Hours a Month Beats a Full-Time Hire** — Who you hire to run this program before you can afford a full-time CISO.
- **Migrating 225K Users from AWS Cognito to Auth0 Without Forcing a Single Logout** — A real-world identity migration at fintech scale — Layer 1 of the stack done right.

Canonical web version: sublimecoding.com/blog/running-security-at-an-ai-native-company-2026

CHAPTER 4

Chapter 4 — What 90 Days of a Fractional Security Engagement Actually Looks Like

> The previous three chapters describe the strategic decisions. This chapter compresses them into operational reality — a sanitized week-by-week composite of a typical 90-day fractional CISO engagement at a Series-Seed AI startup with two enterprise prospects in pipeline. What gets done, what it costs, what graduates the founder out of needing one. If you've read the rest of the playbook and are at "okay, what would the next 90 days actually look like," this chapter is the answer.

Most founders who book the intro call have already read three or four of my posts and arrive at the same question: "Okay, but what would the next 90 days actually look like if I hired you?" Here's the answer — a sanitized week-by-week composite of a typical 90-day fractional security engagement, with the real numbers attached.

Not every engagement looks exactly like this. But this is what most pre-Series-A AI engagements look like — same shape, different details. Composited from real work, no specific client.

The starting state: a Series-Seed AI startup, eight engineers, ARR in the low seven figures, two enterprise prospects in pipeline. Both prospects are stuck at the security questionnaire. One explicitly asked for SOC 2 Type I; the other implied it. The founders haven't run a security program before. The CTO has been triaging the questionnaires personally and hates every minute of it.

The ask: get them through both questionnaires within 90 days, run the SOC 2 Type I audit in parallel, and stand up the security work that needs to outlast the engagement.

Week 0 — the scoping call

Thirty minutes. Three questions. What's the named deal blocked behind this work, who internally owns security after I'm gone, and what's the budget envelope. If any of those answers are squishy, we don't sign. Real engagements don't survive ambiguity at the start.

The answers I'm looking for: a specific deal name, a specific internal owner (usually the CTO or a senior engineer who will absorb the role), and a number — a real one, not a range. Founders who can't answer all three aren't ready for fractional work. They're ready for the conversation about what would have to be true before they were.

Pricing gets confirmed on this call too. Standard fractional CISO retainer is $3K/month for 10–15 hours of strategic work. Audit-prep engagements run four months, occasionally six. Deliverables and out-of-scope items go into a one-page scope doc the next day. No surprise overages. No retainer creep.

Weeks 1–2 — discovery and the policy pack

Discovery is short by design. I'm not running a six-week assessment — I'm running a one-week one because I already know what most pre-Series-A AI startups look like and what they're missing. The week is for confirming the gaps, not for finding them.

What gets done in week one: read all the existing security documentation (usually a one-page README and a handful of Notion pages), interview the CTO and the founder for an hour each, walk the production environment with whoever's on-call, and pull the existing controls into a SOC 2 readiness matrix. By the end of week one, the gap list is on paper.

Week two is the policy pack. Acceptable use, access control, data classification, incident response, vendor management, change management, vulnerability management, business continuity. Eight policies, drafted in the company's voice from templates I've been carrying for years, customized to the actual technology stack. The CTO reviews and signs off. We push them into the Vanta or Drata instance the same week.

By end of week two, the company has policy language that holds up to an auditor's read and a written gap list ranked by audit-blocking severity.

Weeks 3–6 — SOC 2 readiness in parallel with the AI security stack

This is the heavy stretch. Two work streams running in parallel.

The SOC 2 stream: implement controls against the gap list, configure the compliance tool to track evidence collection, schedule the auditor (early — the good ones book out four to six weeks), document the technical controls (MFA enforcement, access provisioning, secure SDLC), and start collecting evidence as the controls go live. Most of the engineering team's involvement happens in week three when we configure the access provisioning tooling and again in week five when we wire the production change management process into their existing PR workflow.

The AI security stream: this is where the work that's not in the SOC 2 framework lives. The four-layer AI-native stack — prompt injection defense, agent credential scoping, secrets handling, audit logging — gets stood up in this window. Credential scoping comes first, because that's where the largest blast-radius incidents originate. By end of week four, every agent and every service account has the minimum credential surface needed to do its job. No "admin to be safe" anywhere.

By end of week six, both streams have visible momentum. Vanta or Drata shows green on most of the framework. The AI-native layer has credential scoping done and prompt injection guardrails in design. Both enterprise prospects get an updated security questionnaire response that shows real evidence of the work — and both move forward on their evaluation.

Weeks 7–10 — auditor engagement and the AI-native layers

Weeks seven and eight are auditor fieldwork. The auditor runs interviews with the team, walks the production controls, samples the evidence, and asks the questions that don't have clean answers. My job in this stretch is to be the security executive in the room with the auditor — answering the technical questions, defending the design decisions, and protecting the engineering team from the death-by-questionnaire pattern that kills first-time SOC 2 attempts.

Weeks nine and ten close out the AI-native layers. Secrets handling moves from ad-hoc (env vars and the occasional `.env` checked into a private branch) to deliberate (Vault or AWS Secrets Manager, automated rotation on the secrets that actually matter). Audit logging gets stood up — every agent action, every privileged service account call, every credential-bearing API request goes to a central log with retention long enough to investigate an incident a month after it happens.

By end of week ten, the auditor's draft report is in review. Both enterprise prospects have what they need. The technical work outlives the engagement.

Weeks 11–12 — report and graduation

Week eleven is the auditor report cycle. Their draft, our review, their final. The Type I report ships at end of week eleven or early week twelve. Both enterprise prospects close their security review within ten business days of the report landing.

Week twelve is the graduation conversation. I always have the same conversation around this point. "You don't need me on the day-to-day anymore. Here's what comes next, and here's when you'd hire someone full-time."

Most engagements either step down to a low-touch advisory retainer — four to six hours per month at $2K/month, mostly for security review questions and the next year's audit prep — or graduate completely. The bad outcome is the one where I'm still the on-call security executive at month nine. That means I haven't built the program right. The internal owner identified in week zero needs to be operational by week twelve, or I've failed the engagement.

What it cost

Total 90-day spend, all in:

- Auditor fee: $15–20K (varies by auditor, scope, and audit-prep tooling)
- SOC 2 tooling (Vanta (https://www.vanta.com) or Drata (https://drata.com), year one): $5–10K
- Fractional CISO retainer: $12K (four months at $3K)
- Internal time: roughly four hours per week of the CTO's attention; less for the engineering team after week three

Total external spend: $32–42K. Inside the $25–45K range I quote in the SOC 2 post, on the higher end because of the AI-native layers added in parallel.

Cost of NOT running this engagement: two enterprise deals that don't close. Pipeline that ages out. The CTO answering security questionnaires personally for the next six months instead of building the product. Easy math.

What actually changed

Two named enterprise deals unblocked. SOC 2 Type I report in hand for the next six prospects. A security program that outlives the engagement: one internal owner, one accountable executive, weekly review cadence, and a roadmap to Type II audit at the next renewal.

Specific operational outcomes: every service account has minimum-viable credentials. Every agent has bounded tool access. Every secret is in a manager with rotation policy. Every privileged action lands in a searchable audit log. None of these existed at week zero.

Cultural outcomes that matter just as much: the CTO knows what to say in a security questionnaire and what to escalate. The engineering team has a security review pattern they can run for new features without me. The founder has a number — a real one — for what compliance costs at the next stage.

When this engagement isn't right

Not every founder should hire a fractional CISO. The wrong fit produces a worse outcome than no fit at all.

Wrong-fit signals:

- The deal blocking on the security review isn't real. ("We think SOC 2 would help us close more enterprise" without a specific named prospect = wrong stage.)
- No internal owner. If nobody on the team will absorb the role at month four, the engagement either extends indefinitely or rolls back inside two quarters.
- Regulated industry that requires a full-time CISO on the org chart for the deals being pursued. Healthcare with PHI, payments, government.

- Security work consistently exceeds 20 hours per week. Past the fractional break-even — full-time is now the right answer. (See the vCISO math post for the graduation criteria.)

If any of those describe you, the conversation we should have isn't about hiring me — it's about what your real next move actually is. I'll say so on the call.

How to get this conversation started

If you're a pre-Series-A AI founder with a named enterprise deal blocked behind security review, the intro call is the right step. Thirty minutes. Three questions. We figure out together whether this engagement shape is the right one for you — and if it isn't, I'll point you at what is.

Engagement model and the next step are here.

Read this next

- vCISO Math for AI Founders — the make-vs-buy argument and the graduation criteria.
- SOC 2 Is a Revenue Tool, Not a Security Tool — the reframing that makes the engagement above worth running.
- How I'd Run Security at an AI-Native Company in 2026 — the four-layer stack the engagement above stands up.

Canonical web version: sublimecoding.com/blog/what-a-fractional-security-engagement-actually-looks-like

How to use this playbook from here

If you've read this far, you have one of three reactions: this isn't relevant to you (no problem, archive the PDF), this is relevant but not yet (bookmark it for the quarter the named enterprise deal lands), or this is the conversation you should be having now.

For the third group: the intro call is the right step. Thirty minutes. Three questions: what's the named deal blocked behind this work, who internally owns security after I'm gone, and what's the budget envelope. We figure out together whether a fractional engagement is the right shape for you — and if it isn't, I'll point you at what is.

Engagement model and the booking link are at sublimecoding.com/consulting (https://sublimecoding.com/consulting).

This playbook is a snapshot. Numbers (SOC 2 costs, fractional CISO retainers, graduation thresholds) reflect 2025–2026 engagement realities and will drift over time. The web versions of each chapter get edited as the work changes; consult the canonical URLs at the start of each chapter for the current take.