A FIELD GUIDE FOR FOUNDERS AND ENGINEERING LEADERS

The Enterprise Readiness Playbook

How security questionnaires, SOC 2 and ISO 27001 controls, and scanner findings actually connect. And the ninety day sequence from reacting to ready.

By **Ranjit Victor**
Built the vulnerability management function. Twenty years in software engineering and leadership.

ranjitvictor.com · ranjitvictor@gmail.com

CONTENTS

What is in here

This is written for the person who has to answer for security but was hired to ship product. No auditor vocabulary, no fear selling. Read it in twenty minutes and you will know where you stand.

> WHO THIS IS FOR
>
> Software companies with roughly ten to eighty engineers, selling to enterprise customers, with no dedicated security hire. If you have a security team already, most of this will be familiar.

01

Why security arrives as a sales problem

Almost nobody starts a security program because security felt important. They start because a deal stopped moving.

The pattern is consistent. You win a large customer. Their procurement team sends a security questionnaire with two hundred questions. Or they ask for your SOC 2 report. Or their security reviewer wants to know your vulnerability remediation SLAs. Suddenly a deal you had already forecast is sitting behind a document you cannot produce.

What follows is usually a lost week. Someone senior stops building and starts guessing at answers. Some answers are optimistic. A few are wrong, not dishonestly, but because nobody actually knows whether the thing being asked about is true. The deal closes eventually, or it does not, and the company goes back to shipping until the next customer asks.

That cycle is expensive in a way that does not show up on any dashboard: senior engineering time consumed at random intervals, sales cycles extended by weeks, and a quiet ceiling on the size of customer you can serve.

> THE REFRAME
>
> Enterprise readiness is not a security project. It is a sales enablement project that happens to be built out of security controls. Treating it that way changes who cares about it internally, and it changes what you build first: the things a reviewer asks about, in the order they ask.

The good news is that the questions are not infinite. Enterprise reviewers and auditors converge on a fairly small set of expectations, and most of them are things a competent engineering team half does already, just without the evidence to prove it.

02

The three languages, and the one idea that translates them

Three groups talk about security and none of them use the same words. Once you see the translation, the whole subject collapses to a manageable size.

Language one: the framework

SOC 2, ISO 27001 and every enterprise questionnaire are written as **controls**. A control is a statement of intent, shaped like this: the organisation identifies, evaluates and remediates technical vulnerabilities in a timely manner. Notice what is absent. No tool names. No CVE numbers. No severity thresholds. Frameworks describe outcomes, never implementations.

Language two: the tool

Your scanners speak in findings. A dependency has a known vulnerability. A storage bucket allows public reads. A credential appears in git history. Findings are facts about your systems at a moment in time. They are inputs, not conclusions, and on their own they satisfy nothing.

Language three: the auditor and the reviewer

Both ask the same three questions about every control, and this is the idea that translates everything:

THEY ASK FOR	WHICH MEANS	WHAT IT LOOKS LIKE
Policy	You have written down what you intend to do	A short document stating that critical vulnerabilities are fixed within a defined window
Process	The thing actually happens, repeatably, by someone	Scans run on a schedule, findings get owners, fixes get tracked
Evidence	You can prove it happened, over time, without preparing specially	Scan history, tickets closed within SLA, access review records

> THE WHOLE TRANSLATION IN ONE LINE
>
> A scanner report is not compliance. A scanner report, plus a written SLA, plus tickets showing findings closed within that SLA, is the complete evidence package for the vulnerability management control. Tools produce the third column only. The first two columns are where almost every company at your stage is missing.

This is why buying more tooling rarely moves the needle. Adding a second scanner triples your findings and changes your control posture not at all. Writing one page of policy and enforcing one SLA changes it completely.

03

The mapping: what a control wants, what a scanner sees

Here is the translation table. Keep it beside you the next time a questionnaire arrives.

Controls where scanner output is the primary evidence

CONTROL AREA	WHAT IS ACTUALLY BEING ASKED	WHAT PRODUCES THE EVIDENCE
Vulnerability management SOC 2 CC7.1 / ISO A.8.8	Vulnerabilities are found, judged and fixed on a clock	Dependency and code scanning, plus your severity SLAs and closed tickets
Secure configuration CC6.6, CC6.8 / A.8.9	Systems are hardened and drift is caught	Cloud posture scanning: public buckets, open security groups, defaults left on
Encryption CC6.7 / A.8.24	Data is protected in transit and at rest	Posture findings: unencrypted volumes, weak TLS, missing key management
Secrets handling CC6.1 / A.8.28	Credentials are not lying around	Secret scanning across full git history, ideally at commit time as well
Network protection CC6.6 / A.8.20 to A.8.22	Networks are segmented and exposure is deliberate	Posture findings on open ports, permissive rules, flat networks

Controls where the scanner only tells half the story

For these, a tool shows you the configuration but a person has to confirm the practice. These are the questions a reviewer will ask that no dashboard answers.

CONTROL AREA	THE TOOL HALF	THE HALF YOU HAVE TO ANSWER YOURSELF
Access control CC6.1 to CC6.3 / A.5.15	Over privileged roles, unused keys, accounts without MFA	Is there a joiner and leaver process? Who approves access? When were permissions last reviewed?
Change management CC8.1 / A.8.32	Pipeline configuration is inspectable	Are reviews mandatory? Can anyone push to production? How do you roll back?
Logging and monitoring CC7.2, CC7.3 / A.8.15	Whether audit logging is switched on and retained	Does anyone look at it? Where do alerts go at 2am, and who answers?
Incident response CC7.4, CC7.5 / A.5.24	Detection tooling exists	Is there a written plan with severity levels? Has it ever been tested? Do you write post incident reviews?
Backup and recovery A.8.13 / Availability	Backup configuration exists	Has a restore actually been tested? Do you know your recovery time objective?

> THE SINGLE MOST USEFUL QUESTION
>
> For any control you are unsure about, ask: **who does this, how often, and where is the record?** If there is no answer, it is a gap, regardless of framework or tooling. That one question is essentially the entire audit profession compressed into nine words.

04

The three thirds of any framework

SOC 2 has around sixty relevant criteria. ISO 27001 Annex A has ninety three controls. Both numbers are intimidating until you see that they split cleanly into three groups with completely different owners.

The technical third

Vulnerability management, configuration, encryption, secrets, network security, secure development. This is engineering work, it is where scanners live, and it is the part enterprise reviewers probe hardest at software companies, because it is the part most likely to be theatre. It is also the only third that genuinely requires engineering judgment.

The process third

Access reviews, change management, incident response, monitoring practice, backup testing. These wrap around technical systems but are fundamentally about human routine. Most companies at your stage do some of this informally and have no record of it. Converting informal habits into recorded routine is usually two weeks of work, not two months.

The paperwork third

Policy documents, risk assessment, vendor management, HR controls such as background checks and security training, asset inventory, data classification, business continuity. This is the part that feels alien to engineers, and it is the part that compliance automation platforms have turned into fill in the blanks workflows. Your operations or HR function owns most of it.

> WHAT THIS MEANS PRACTICALLY
>
> You do not need a compliance expert to start. You need someone to make the technical third true, someone to turn the process third into routine, and a platform plus your ops team for the paperwork third. Companies stall because they try to do all three at once, in framework order, instead of in risk order.

A note on India

If you handle personal data of Indian users, the Digital Personal Data Protection Act adds obligations around consent, breach notification and data principal rights. It sits alongside SOC 2 or ISO rather than replacing them, and most of the technical controls you build for one will serve the other.

05

Why your severity scores are lying to you

Run a scanner across a typical codebase and cloud account and you will get somewhere between several hundred and several thousand findings, a large share of them marked high or critical. If you try to fix by severity, you will fail, and worse, you will lose your team's trust in the process.

The reason is that the standard severity score, CVSS, measures how bad a vulnerability would be if exploited, in the abstract. It says nothing about whether anyone is actually exploiting it, and nothing about whether the affected thing is reachable in your environment. A critical score on a library used by an internal batch job that runs once a month is not an emergency. A medium score on your internet facing authentication path might be.

The three signals that actually rank risk

SIGNAL	WHAT IT TELLS YOU	WHERE IT COMES FROM
Known exploitation	Attackers are using this right now, in the real world. This overrides everything else.	The CISA Known Exploited Vulnerabilities catalogue, published and free
Exploit probability	The statistical likelihood this gets exploited in the next thirty days, expressed as a percentage	EPSS, published by FIRST, updated daily and free
Your exposure	Is the affected thing internet facing, and does it touch customer data	Only you know this. It comes from your architecture, not from any tool.

Combine those three and the picture inverts dramatically. In a typical assessment, a four figure pile of findings resolves to a single figure list of things that genuinely deserve a sprint. The rest are real, worth fixing eventually, and not worth interrupting a roadmap for.

> THE PRACTICAL TEST
>
> If your security process cannot tell an engineer which five things to fix first, it is not a process, it is a backlog. And backlogs that nobody can triage get ignored, which is how companies end up with scanning switched on and nothing improving for a year.

One exception worth knowing

A credential found in source control that is confirmed still active is a maximum priority item regardless of any score, because there is no exploitation to predict. The exposure has already happened. Rotate first, investigate afterwards.

06

The ninety day sequence

This is the order that works, and the order matters more than the speed. Each phase produces something you could show a customer at the end of it.

DAYS 1 TO 15 · SEE CLEARLY

Establish the baseline

You cannot prioritise what you cannot see, and you cannot answer a questionnaire honestly without this.

- Scan everything once: dependencies, code, containers, secrets across full git history, infrastructure as code, and cloud posture across all accounts and regions
- Review identity: who has access to what, who has administrative rights, is multi factor enforced, are departed people still enabled
- Map exposure: which services are internet facing, which handle customer data, where do the production boundaries sit
- Rank the findings by known exploitation, exploit probability and exposure. Produce a shortlist, not a spreadsheet
- Write down who owns which service. In a small company the honest answer may be one person, and that is itself a finding

DAYS 16 TO 45 · FIX AND INSTRUMENT

Close the real gaps, then make scanning permanent

- Fix the shortlist. Rotate any live credentials immediately, close public exposure, remove wildcard administrative permissions
- Wire scanning into your pipeline so it runs on every change rather than on demand. Tune it hard: an alert nobody trusts is worse than no alert
- Add secret detection at commit time so the next credential never reaches history
- Set severity based response windows and write them down. One page. Include an exception path, because you will need one, and an exception that is recorded is not a failure
- Switch on your cloud provider's own security services. Most companies are already paying for capability they have never enabled

DAYS 46 TO 90 · MAKE IT PROVABLE

Turn practice into evidence

- Stand up a compliance platform if a formal audit is coming, and connect it to your systems so evidence collects itself
- Run your first access review and record it. Quarterly from then on
- Write the short policy set: information security, access control, incident response, vendor management. Short and true beats long and aspirational
- Write and walk through an incident response plan with severity definitions and named responders
- Test a restore from backup. Write down what happened, including if it went badly
- Answer your standard questionnaire once, properly, and keep the answers. The next one becomes an editing exercise

> WHAT YOU HAVE AT DAY 90
>
> A ranked and shrinking risk list with owners. Scanning that runs itself. Response windows that are met. Evidence that accumulates without anyone preparing it. And a questionnaire you can answer in an afternoon, truthfully.

07

Answering a questionnaire without losing a week

Security questionnaires feel adversarial. They are not. The reviewer on the other side is trying to decide whether you are a risk to their company, and they have seen hundreds of responses. Here is what they are actually reading for.

They are testing consistency, not perfection

No reviewer expects a fifty person company to have the controls of a bank. They expect your answers to hang together. If you claim quarterly access reviews and cannot produce one, that single inconsistency does more damage than admitting you do them annually.

Never answer yes to something you cannot show

The honest formulation is powerful and widely accepted: describe what you do today, then state what is planned and when. In practice, a reviewer will accept a documented gap with a date far more readily than a claim that collapses under one follow up question.

Keep a written answer library

Most questionnaires overlap by seventy percent or more. Answer once, keep the answers with links to the evidence, and review them quarterly. This is the single highest return administrative habit in the whole subject: it converts a recurring week long fire into an afternoon.

Push back on scope where it does not apply

Questionnaires are generic and will ask about physical data centre controls, mainframes, or practices irrelevant to a cloud native company. Not applicable, with one line of reasoning, is a perfectly professional answer and reviewers see it constantly.

> THE PHRASE THAT MATTERS
>
> A common and costly mistake is promising to be certified. Certification comes from an accredited auditor or certification body, not from you and not from a consultant. What you can commit to is being audit ready, and knowing the difference is itself a signal to a reviewer that you understand the process.

08

Five traps that cost companies months

Buying tools before defining a process

A scanner without an owner, an SLA and a triage rule produces findings nobody acts on, and quietly teaches your team that security alerts are noise. Fix the process first. The tools are the easy part and several capable ones cost nothing.

Chasing the framework instead of the risk

Working through a control list top to bottom means spending week one on documentation while a live credential sits in your git history. Do the risk work first, then map what you built to the framework. The mapping is far easier in that direction.

Treating compliance and security as the same thing

They overlap, they are not identical. A company can be audit ready and still be soft in ways an attacker would enjoy. Certification is a floor and a sales unlock, never a conclusion.

Starting three months before the deadline

A Type II report observes your controls operating over a period, commonly three to twelve months. You cannot compress that window. If an enterprise customer is on your roadmap for next year, the practice needs to start running this year, because the evidence has to accumulate in real time.

Making it one person's side project

Security assigned to whoever is least busy stalls the moment that person gets busy, which is immediately. It needs either a named owner with protected time, or outside help with a fixed scope and a deadline. Diffuse ownership is the most common reason programs die at day forty.

09

Where do you stand

Ten questions. Answer them honestly with your engineering lead. Any answer that is not a clear yes with evidence behind it is a gap, and gaps are ordinary. The count tells you how much runway you need before an enterprise reviewer arrives.

- ☐ Do you know, today, how many known vulnerabilities exist in your dependencies and cloud configuration?
- ☐ Do scans run automatically on every change, rather than when someone remembers?
- ☐ Is there a written window within which a critical issue must be fixed, and was it met last quarter?
- ☐ Has anyone scanned your full git history for credentials, including deleted branches?
- ☐ Can you name, right now, every person with administrative access to production and confirm they still need it?
- ☐ Is multi factor authentication enforced everywhere, with no service account exceptions?
- ☐ If a customer asked for proof of your last three months of remediation activity, could you produce it without preparing anything?
- ☐ Is there a written incident response plan, and has anyone walked through it?
- ☐ Has a restore from backup been tested in the last six months?
- ☐ Could you answer a two hundred question security questionnaire this week without stopping feature work?

READING YOUR SCORE

Eight or more clear yes answers. You are in good shape. Focus on evidence and documentation rather than new controls.

Four to seven. Typical for a growing company. There is real work here, and ninety days of focused effort will get you there.

Three or fewer. Also typical, and not a crisis. But do the baseline scan soon, because the unknown findings are the ones that hurt, and an enterprise deal will force this timeline for you.

About the author

Ranjit Victor spent three years building and running the vulnerability management function at a global SaaS integration platform: the scanning pipelines, the prioritisation model, the response SLAs, the build time security gates and the executive reporting. Before that, twenty years in software engineering and leadership, including a decade at Misys building enterprise banking software and eight years consulting for organisations including Societe Generale, MetricStream and Tala Mobile.

He now helps growing software companies pass enterprise security scrutiny, starting with a three day Security Snapshot that requires no tooling and no purchase on your side.

Find out where you stand. Thirty minutes, no charge. Bring your last security questionnaire and see what it was really asking.
ranjitvictor@gmail.com · ranjitvictor.com · linkedin.com/in/ranjitvictor