

THE PLAYBOOK

Secure by default

The principles behind how I build software — why security is a design constraint from the first line of code, and what that discipline protects you from.

THE PRINCIPLE

Security is a constraint from the **first line of code** — never a retrofit.

Bolted-on security is where breaches live. When protection is an afterthought, the gaps are already shipped by the time anyone looks. So I design it in: the rules below govern the build before a feature exists, and an adversarial review proves them before anything goes live.

This matters more, not less, in the age of AI-assisted development — which can move fast and confidently introduce real risks: invented dependencies, permissive defaults, and endpoints wide open to abuse. The discipline is what keeps speed from becoming exposure.

01 Assume it's already broken

Every build is reviewed like a hostile penetration test that assumes at least three critical mistakes are hiding in it — and goes looking until it proves otherwise. Confidence is earned adversarially, not asserted.

02 Trust nothing from the browser

The client can be tampered with, so it never gets the final say. Every permission and validation decision is enforced again on the server, which returns only what the user is entitled to.

03 Least privilege, by default

Data is locked down and opened deliberately — never the reverse. Row-level security means each client and user can physically reach only their own slice, even if something upstream slips.

04 Secrets never live in code

Keys and credentials are stored as environment secrets and read only at runtime — never hardcoded, never committed. The repository can be shared without handing over the keys.

05 Every dependency earns its place

Third-party packages are verified as real and reputable before they're trusted — a deliberate guard against the fake or malicious packages that fast, AI-assisted builds can pull in.

06 Fail closed, fail friendly

When something breaks, the safe path wins: no data leaks, no raw errors, no blank screens — just a clean, on-brand page while the real detail is logged where only the team can see it.

07 Nothing ships without a PASS

The gate is not optional. A build deploys only after the review returns a clean verdict — or after a specific risk is consciously accepted, documented, and justified in writing.

How the principles become a process

The beliefs above aren't a poster on the wall — they run as four concrete stages, with a single hard bar between building and shipping.

01 • PRE-BUILD Design it in Standing rules set before code: validation, server-side authorization, secrets, least-privilege data, hardened headers, resilient UX.	02 • THE GATE Prove it An adversarial audit assumes mistakes exist, hunts them across every axis, and fixes each with an exact change.	03 • DEPLOY Ship it safely Back up, deploy, confirm health, keep rollback ready. Guardrails block wrong-target overwrites.	04 • POST-LAUNCH Watch it Error capture, health checks, and abuse monitoring from day one, with IP-ban on standby.
---	--	--	--

THE HARD BAR

Between stage 2 and stage 3 is a PASS/BLOCK gate. A Critical finding, or an unresolved High, blocks the release. The verdict and its reasoning are written into a per-build security record — including any risk consciously accepted, with its scope and justification. Decisions are documented, never assumed.

Consistent across everything I build

This is one standard, applied the same way to every client dashboard and internal tool — so "secure" means the same thing on your project as on every other.

THE STANDARD, EVERY BUILD

Speed you can move at, without exposure you can't afford.

Good security shouldn't slow you down — done right, it's what lets you ship quickly and sleep at night. That's the whole point of a repeatable playbook: the rigor is built in, so you get the speed and the safety.

HONEST SCOPE

This is a rigorous engineering practice, not a formal HIPAA / SOC 2 certification unless separately scoped. If you need a formal compliance program, we'll plan it deliberately.

LET'S TALK ABOUT YOUR BUILD

hello@thecvmethod.com

Strategy that ships.