FREE CHECKLIST FROM THE TALK "VIBE CODED AND VULNERABLE"

The 15 things to check in AI-generated code *before you ship it*

Run them in this order. Most take minutes. None need a security background, just two test accounts and a fresh AI session. The prompts on page 2 are ready to paste into Claude Code, Cursor or Copilot.

BEFORE THE FIRST PROMPT

- [ ] 01 **Security rules live in your rules file.**
  CLAUDE.md, .cursor/rules, copilot-instructions.md or AGENTS.md. Template on page 2. Ten minutes, once.

WHO CAN DO WHAT

- [ ] 02 **Every API route and server action checks the session on the server.**
  Not in the page, not in the button. On the server.
- [ ] 03 **Every route that takes an ID checks the owner.**
  "Logged in" is not "allowed". Sign in as user B and open user A's links. You should get "not found".
- [ ] 04 **Admin actions check the role on the server.**
  Hiding the Refund or Delete button is decoration, not access control.

YOUR DATABASE

- [ ] 05 **Supabase: row level security is on for every table, each with a policy.**
  Run the Security Advisor in the dashboard. New table means RLS and a policy in the same migration.
- [ ] 06 **Firebase: no rule says `if true`.**
  Test mode expires after 30 days. The fix is real rules, not opening everything.
- [ ] 07 **Test every policy as a second user.**
  A policy you have not tested is a guess.

SECRETS

- [ ] 08 **No secret keys in client code or public env vars.**
  Anything named NEXT_PUBLIC_ or VITE_ ships to every visitor. Search your built JS for "sk_" and "secret".
- [ ] 09 **.env has never been committed.**
  Check git history, not just today's files (gitleaks does it free). If a secret ever leaked: rotate it. Deleting is not enough.
- [ ] 10 **Production does not serve source maps or staging to the public.**
  Your full source code should not be one click away.

HOW YOU BUILD WITH AI

- [ ] 11 **Every new package is real and maintained.**
  Models invent package names, and people register them. Check the registry page before you install.
- [ ] 12 **Agents never hold production keys, and MCP servers are vetted.**
  Treat an MCP server like a browser extension from a stranger. Never paste customer data or .env files into a prompt.
- [ ] 13 **If your product has AI features: tools are narrow and scoped by the server.**
  The system prompt is a request, not a lock. Give the model buttons, not keys, and treat its output as untrusted input.

REVIEW AND SHIP

- [ ] 14 **A fresh AI session reviews every change, and each finding gets a failing test.**
  /security-review or the attacker brief. Never let the chat that wrote the code review it. No test, no finding.
- [ ] 15 **CI is green before release: AI review, secret scanning, package alerts.**
  Plus a one-page incident plan. Under the NDPA, a personal-data breach goes to the NDPC within 72 hours.

Prompts to paste, *in a fresh session*

Never in the chat that wrote the code. For anything they find, ask for a failing test before you ask for a fix.

The attacker brief (the full review)

```
You are reviewing this app as an attacker would. Do not fix anything yet.
For every API route and server action, list:
  1. who can call it, and what it checks
  2. how that check could be bypassed, in one sentence
  3. severity, file and line
Only report what you can show with a real request, against a local or staging copy.
Then check: RLS on every table, secrets in client code,
.env in git history, admin features hidden only in the UI.
```

One prompt per hole

```
1. List every server route that relies on the UI to hide it, with file and line.
2. Find secrets that reach client code or public env vars (NEXT_PUBLIC_, VITE_).
3. For each table, show its RLS policy or security rule, or say it has none.
4. For each route that takes an ID, show the line that checks the owner, or say it is missing.
5. Search git history and /public for keys, tokens and .env files.
```

Prove it

```
For finding N, write a test that fails today because of it and will pass once it is fixed.
Use two test users. Do not change application code yet.
```

Rules file template (CLAUDE.md / AGENTS.md)

```
## Security rules: never break these
- Every API route checks session AND owner.
- Hiding a button is not access control.
- No secrets in NEXT_PUBLIC_ or VITE_ vars.
- New table = RLS + policy, same migration.
- Never commit .env*. Never weaken .gitignore.
- New package? Confirm it is real and active.
- Each fix ships with a test that proves it.
```

Free tools to switch on

```
Claude Code     /security-review
GitHub Action   anthropics/claude-code-security-
review
Secrets         gitleaks
Packages        Dependabot, OSV-Scanner
Supabase        Security Advisor
Others          Cursor Bugbot, Copilot review,
                Semgrep, Snyk
```


Free to share. Test only what you own or are allowed to test.