

FIELD GUIDE 01 / REVISED EDITION

Vibe Coding Security Launch Kit

A practical routine for checking an AI-built app before real users, private data and payments arrive.

What you get

A pre-launch checklist, platform examples, copy-and-paste prompts and evidence worksheets. Use Vibe App Scanner for the deployed checks it can reach, then prove the boundaries that still need manual testing.

A working demo is the starting point

The important question is what happens outside the intended workflow: a different account, a changed object ID, an old session, a repeated payment event or an untrusted file.

Built for founders and developers using Lovable, Bolt, Replit, Base44, Cursor, Claude Code, Codex and other modern web stacks. Supabase and Firebase examples are included.

JACOB PERKS / THUNKLE

Cybersecurity practitioner and software builder. Built and sold SecurEcommerce; founder of Vibe App Scanner.

Version 2.0 / 20 October 2026. Educational material, not an audit, certification or security guarantee.

Free sample / three selected pages

Coverage map + safe two-account test setup. No email required.

The full 34-page kit is \$29 USD. The VAS bonus is for paid buyers.

AUTOMATED OBSERVATIONS / MANUAL EVIDENCE

A / What VAS covers, and what it cannot prove

VAS tests a running app. It does not automatically see every route, account, database rule or business workflow. Treat this as a coverage guide, not a promise that each check completed on your scan.

Area	VAS can contribute	You or a developer must still prove
Secrets & exposure	Inspect discovered pages, bundles, exposed files and API responses.	Private repository history, CI output, server-only credentials and whether exposed keys were rotated.
Database & APIs	Probe discovered backend surfaces and accessible endpoints for exposure.	Effective grants, all policies, protected fields and every role's intended permissions.
Login & access	Evaluate reachable authentication surfaces; configured authenticated tests add coverage.	Recovery, deactivation, tenant rules, invitations and the full A/B account matrix.
Headers & client	Inspect delivered headers, TLS, client patterns and known front-end dependency versions.	A complete dependency inventory, exploitability and application-specific CSRF/XSS controls.
Payments & AI	Identify exposed payment credentials and visible AI/LLM configuration risks.	Entitlements, refunds, replay, concurrent jobs, tool permissions and spending limits.
Recovery & operations	Provide evidence and remediation guidance for observed findings.	Restore a backup, verify alerts, rehearse incidents and review production access.

Check the scan's reach before the score

Confirm the target, time, scan type, URLs reached, test account and incomplete/skipped checks. A timeout, login wall, blocked request or undiscovered endpoint is unknown, not a pass. A clean scan means no findings in the completed checks, not that the whole app is secure.

The included Go month provides standard scans and full findings. Pro adds deeper crawling and more extensive authenticated access-control testing when configured. Neither replaces a developer's review. Repository scanning is advertised as coming soon as of 20 October 2026; do not count it as completed coverage.

Coverage basis: [VAS security checks](#) and [current plan descriptions](#), reviewed 20 October 2026.

SAFE SCOPE / POSITIVE CONTROLS / REPEATABLE EVIDENCE

B / Set up a test you can trust

Use a staging environment that represents production, or explicitly agree a narrow production test scope. Use synthetic records, an email sandbox and payment-provider test mode. Do not test a third party's app or create real charges, mail or destructive changes as a shortcut.

1 / Define the boundary

Write the permitted hosts, test window and allowed actions. Record the deployment or commit, schema version, account roles and integrations. Decide who can stop the test and how test data will be removed.

2 / Create separate identities

Use ordinary accounts A and B in different workspaces, plus an invited admin if needed. Start with no shared memberships. Create one synthetic project and private file owned by each account.

3 / Prove the feature works first

As B, confirm B can read and edit B's project. Save its starting state. If the normal action already fails, a failed cross-account attempt does not prove authorization is correct.

4 / Change only the attacker-controlled input

As A, repeat the equivalent request for B's test object. Keep A's own session. Try read, list, update, download and export separately. Do not use an admin key to simulate an ordinary user.

5 / Verify the actual effect

Check the response body and B's stored state. An error page, hidden button, 403 or empty UI is not sufficient if the backend changed a row or exposed data before returning.

Keep the evidence safe

Record method, route, role, expected result and redacted outcome. Store tokens outside screenshots and shared reports. Never paste cookies, service-role keys, full customer records or production secrets into an AI chat.

Starter test: B can read B's object; A and signed-out callers cannot read it, list it or change it. Repeat for files, background jobs and admin functions. Mark anything you cannot test as unresolved.

Continue with the full 34-page kit / \$29 USD

Get platform examples, coding prompts, release worksheets and a month of VAS Go.

Go requires a card; first month \$0, then \$19/month unless cancelled.

thunkle.ai/vibe-coding-security-launch-kit